When Structure Doesn't Help: LLMs Do Not Read Text-Attributed Graphs as Effectively as We Expected

Haotian Xu
Stony Brook University
haotian.xu@stonybrook.edu

Yuning You
California Institute of Technology
ynyou@caltech.edu

Tengfei Ma
Stony Brook University
tengfei.ma@stonybrookmedicine.edu

Abstract

Graphs provide a unified representation of semantic content and relational structure, making them a natural fit for domains such as molecular modeling, citation networks, and social graphs. Meanwhile, large language models (LLMs) have excelled at understanding natural language and integrating cross-modal signals, sparking interest in their potential for graph reasoning. Recent work has explored this by either designing template-based graph templates or using graph neural networks (GNNs) to encode structural information.In this study, we investigate how different strategies for encoding graph structure affect LLM performance on text-attributed graphs. Surprisingly, our systematic experiments reveal that: (i) *LLMs leveraging only node textual descriptions already achieve strong performance across tasks*; and (ii) *most structural encoding strategies offer marginal or even negative gains*. We show that explicit structural priors are often unnecessary and, in some cases, counterproductive when powerful language models are involved. This represents a significant departure from traditional graph learning paradigms and highlights the need to rethink how structure should be represented and utilized in the LLM era. **Our study is to systematically challenge the foundational assumption that structure is inherently beneficial for LLM-based graph reasoning, opening the door to new, semantics-driven approaches for graph learning.**

1 Introduction

Graphs are fundamental data structures for modeling relationships across diverse domains. Their capacity to capture interactions makes them invaluable for both data representation and reasoning. Over the past decade, the machine learning community has widely adopted graphs to unify multimodal data [Dwivedi et al., 2022, McCallum et al., 2000, Sen et al., 2008a], with Graph Neural Networks (GNNs) emerging as the standard approach [Kipf and Welling, 2017, Veličković et al., 2018, Xu et al., 2019, Hamilton et al., 2017, Chen et al., 2018, Wang et al., 2023, Müller et al., 2024, Neubauer et al., 2024, Ying et al., 2021]. Recently, the rise of Large Language Models (LLMs) has opened new opportunities for integrating linguistic reasoning into graph learning, giving rise to graph foundation models.

LLM-GNN hybrids aim to combine the generalization and reasoning abilities of LLMs with the structural inductive biases of GNNs. This integration has shown promise on textual attribute graphs, where nodes carry rich semantic content. Strategies, shown in Figure 1 such as prompt-based graph encoding, hybrid model architectures, and structure-aware instruction tuning have been explored [Chen et al., 2024, Wang et al., 2024, Perozzi et al., 2024, He et al., 2024]. However, the role of structural information in these models remains uncertain. For example, Bechler-Speicher et al. [2024] show

Xu et al., When Structure Doesn't Help: LLMs Do Not Read Text-Attributed Graphs as Effectively as We Expected . *Proceedings of the Fourth Learning on Graphs Conference (LoG 2025)*, PMLR 338, Hybrid Event, December 10–12, 2025.

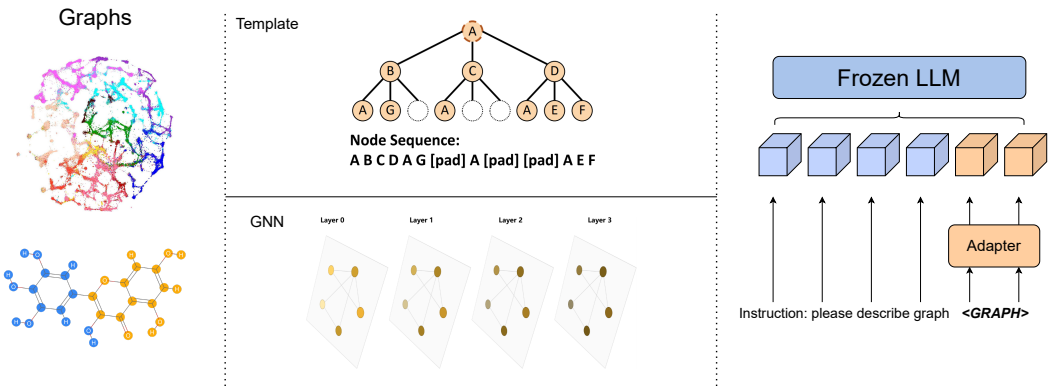


Figure 1: We present a common paradigm for aligning graph type data into LLMs. On the left, one needs to define the graph (citation network, molecule, protein, etc) and parameterize it with proper structures. In the middle, we briefly delineate the strategies encoding graphs into a LLM-favored representations: Template-based encoding will arrange each node inside graph according to a pre-defined sequence, while GNN-based encoding is to have a pretrained or random initialized GNN module to encode graphs into LLM hidden space. On the right is the pipeline to align graph modality into LLMs.

that GNNs may over-rely on structure even when it's irrelevant, while structure-agnostic models like DeepSets [Zaheer et al., 2017] often generalize well. Additionally, standard graph benchmarks may fail to reflect real-world relational complexity, raising concerns about their validity [Bechler-Speicher et al., 2025].

In this work, we take a methodological perspective to re-examine the necessity of structural encodings in LLM-based graph learning. Through systematic experiments across multiple graph types, encoding templates, and modeling paradigms, we find that the inclusion of structural information, whether predefined positional encodings or message passing networks, often yields limited or no performance gains when rich semantic node features are present. In some cases, structural signals can even degrade performance due to oversmoothing or noise. We question the prevailing assumption that graph structure is inherently beneficial and suggest a shift toward more minimal, semantics-centered representations when using LLMs for graph-related tasks. **Our study calls for a rethinking of graph learning in the era of powerful language models, advocating for the design of LLM frameworks that prioritize meaningful textual context over handcrafted structural encodings.** Our code is available at https://github.com/hxu105/llm-graph.

2 Related Work

Graph Learning: Graph learning offers a flexible framework for modeling relational and structural data across domains such as social networks, biology, and knowledge graphs. At the core of this field are Graph Neural Networks (GNNs), which learn node- and graph-level representations through message passing and neighborhood aggregation [Kipf and Welling, 2017, Hamilton et al., 2017]. Variants like Graph Attention Networks [Veličković et al., 2018] and spectral methods [Bruna et al., 2013] have been developed to address limitations in scalability and expressiveness. Inspired by advances in NLP and vision, self-supervised learning has gained popularity in the graph domain, with methods such as GraphCL [You et al., 2020], G-BERT [Shang et al., 2019], and GPT-GNN [Hu et al., 2020] employing contrastive or masked prediction objectives to improve generalization. However, unlike in NLP and vision, graph pretraining lacks standardized benchmarks and consistent input formats, making it harder to transfer models across domains. In response, graph foundation models (GFMs) such as GraphMAE [Hou et al., 2023], GRAND [Feng et al., 2020], and GraphMVP [Liu et al., 2022] aim to learn general-purpose graph representations. Despite their progress, challenges like data heterogeneity and the absence of a shared vocabulary persist—fueling growing interest in leveraging large language models (LLMs) to enhance graph representation learning.

LLMs as GFMs: Recent studies have advanced beyond traditional GNN-based graph foundation models (GFMs) by exploring large language models (LLMs) as graph learners, leveraging their

strong generalization and multimodal capabilities. Fatemi et al. [2024] provides a comprehensive analysis of how graph-to-text encoding strategies influence LLM performance, highlighting the importance of task type, encoding method, and graph structure. Building on this, LLaGA [Chen et al., 2024] introduces a unified framework that transforms graph data into LLM-friendly sequences using structure-aware node reordering and projection, achieving strong generalization and interpretability. PromptGFM [Zhu et al., 2025] integrates in-text graph prompting and a learned graph vocabulary to unify GNNs and LLMs, enabling scalable and transferable reasoning on textual attribute graphs. Ge et al. [2025] improves graph prompting by showing that the sequential order of graph descriptions significantly affects LLM reasoning performance on graph tasks. Similarly, LLM-BP [Wang et al., 2025] enhances inference by combining task-adaptive LLM embeddings with belief propagation guided by LLM-estimated homophily scores. Huang et al. [2024] investigate the role of structural information when incorporated into natural language prompts, while our work focuses on modality alignment and how LLMs internally process graph modality through adapters. Nevertheless, our findings share a similar observation with Huang et al. [2024]: LLMs tend to interpret structure-aware prompts more as contextual narratives rather than explicit topological signals. Wu et al. [2025] introduce *LLMNodeBed*, a benchmark analyzing when LLMs help in node classification across datasets and paradigms. In contrast, our work focuses on how LLMs process graph information, revealing through controlled ablations that they often act as unordered set readers when node semantics dominate, providing a mechanistic understanding rather than a benchmarking study.

In contrast, hybrid approaches like GraphToken [Perozzi et al., 2024] inject structural information via GNN adapters and parameter-efficient prompts. Extensions such as G-Retriever [He et al., 2024] and TEA-GLM [Wang et al., 2024] further integrate structural and textual features to achieve strong performance across graph-text benchmarks. SKETCH [Zhou et al., 2025] fuses graphs with LLMs by embedding structural and semantic aggregation into text encoding; GraphInsight [Cao et al., 2025] mitigates positional bias through strategic placement of key graph information and RAG-style external retrieval to boost structural understanding; GALLa [Zhang et al., 2025] utilizes GNNs to inject code structural information as an auxiliary task. Guan et al. [2025] investigate LLM attention patterns on graph inputs and find that transformer attention fails to align with actual graph connectivity—suggesting a gap in how LLMs internally process structural cues, rather than evaluating their downstream utility. However, most of previous works hold the assumptions that LLMs share the same inductive bias as GNNs, while we question such a belief and assess the role of structural information for LLMs processing graphs.

Table 1: TAG Datasets selected in experiments.

Dataset	Text Domain	Graph Structure
Cora [McCallum et al., 2000]	Publication	Homophilic
Citeseer [Giles et al., 1998]	Publication	Homophilic
Pubmed [Sen et al., 2008b]	Publication	Homophilic
School [Craven et al., 1998]	Webpage	Heterophilic
Roman Empire [Platonov et al., 2023]	Wikipedia	Heterophilic
Amazon Ratings [Platonov et al., 2023]	E-commerce	Heterophilic

3 Do LLMs Read TAG as Expected?

In standard graph learning, models aim to capture relationships between entities by combining semantic information, such as node features or textual descriptions, with structural information derived from graph connectivity. While node attributes provide rich local context, structural links define how entities interact within a broader topology, a dual perspective widely credited for the effectiveness of Graph Neural Networks (GNNs) across many downstream tasks. Motivated by this, recent research integrating Large Language Models (LLMs) with graphs has largely focused on injecting structural signals into LLMs. Parameter-free methods like LLaGA [Chen et al., 2024] verbalize graph structure via handcrafted templates, whereas hybrid approaches such as GraphToken, G-Retriever, and TEA-GLM [Perozzi et al., 2024, He et al., 2024, Wang et al., 2024] employ GNN-based adapters to encode structure into learned embeddings, combining the relational inductive biases of GNNs with the expressive capabilities of LLMs.

Table 2: To evaluate the utility of Laplacian embeddings for LLMs, we compare LLaGA's ND template with our heuristic templates, HN and CO, where HN-1 samples node sequences from the 1-hop neighborhood. As shown below, explicit structural encodings do not consistently enhance performance and can even degrade it in some cases.

Setting	Dataset	Node Classification			Link Prediction	
		ND	HN-1	CO	ND	HN-1
Homophilic	Cora	88.07% (0.74%)	**88.56%** (0.80%)	85.42% (1.78%)	85.56% (1.33%)	**87.27%** (1.56%)
	Citeseer	**80.31%** (0.81%)	80.20% (0.94%)	77.74% (0.31%)	86.73% (0.63%)	**88.79%** (0.84%)
	Pubmed	92.56% (0.71%)	94.80% (0.17%)	**94.84%** (0.04%)	88.25% (0.31%)	**90.98%** (0.38%)
Heterophilic	Shool	66.43% (3.69%)	82.02% (12.79%)	**91.13%** (1.66%)	**68.61%** (0.21%)	68.12% (1.51%)
	Roman Empire	48.56% (1.17%)	59.70% (2.42%)	**62.24%** (0.19%)	81.59% (0.50%)	**83.81%** (0.12%)
	Amazon Ratings	40.97% (0.56%)	**41.67%** (0.22%)	40.38% (1.14%)	80.26% (2.01%)	**84.51%** (0.53%)
Across Datasets		69.48%	74.49%	**75.29%**	81.83%	**83.91%**

These strategies generally fall into two categories: (1) template-based methods that incorporate neighbor aggregation or positional encodings, and (2) GNN-based methods that learn structural representations through neural encoders. Despite their architectural differences, both approaches often yield similar performance. In many text-rich graph tasks, the added structural information, whether hand-made or learned, contributes marginal gains or even degrades performance when strong node-level semantics are already present. This suggests that LLMs may primarily treat input graphs as unordered sets, relying more heavily on the content of selected node sequences than on the underlying graph topology. Our findings challenge the common assumption that structural information is essential for LLM-based graph modeling, and they call for a rethinking of how structure should be incorporated, if at all, into future graph foundation models for semantically rich settings.

3.1 Preliminary

We revisit recent LLM-Graph approaches, such as LLaGA [Chen et al., 2024] and GraphToken [Perozzi et al., 2024], focusing on modality fine-tuned node classification and linke prediction in textual attribute graphs (TAGs). Our analysis is guided by two key questions: (1) Are explicit structural encodings, like Laplacian embeddings, necessary for LLMs? (2) How does message passing networks like GNNs affect performance? We conduct most of our experiments using Vicuna-7b-v1.5 [Zheng et al., 2023].

Datasets As summarized in Table 1, we evaluate our models on six real-world TAG datasets spanning diverse text domains and structural properties. These include citation networks, e-commerce platforms, historical Wikipedia articles, and web page graphs, covering both homophilic and heterophilic patterns. Additional experiment details are provided in Appendix A, B and C.

3.2 Template-Based Encoding

In this subsection, we revisit the LLaGA framework [Chen et al., 2024], with a particular emphasis on the *Neighborhood Detail (ND)* template. This template is built upon a predefined computational graph, typically a k hop B tree, and incorporates Laplacian-based positional encodings to inject structural priors into the LLM input. To rigorously evaluate the contribution of these structural components, we conduct a systematic ablation study in which both the handcrafted subgraph and the positional encodings are removed and replaced with a simple, order-invariant sequence of node descriptions.

We benchmark the original ND template against two lightweight, structure-agnostic variants: (1) **HN (Hop Neighbor)**, which randomly samples a subset of k-hop neighbors to construct the node sequence, and (2) **CO (Center Only)**, which provides only the description of the central node. As shown in Table 2, the ND template fails to surpass the other two structure-free templates in both node classification task and link prediction task. And including such structural embeddings can be harmful for LLMs recognizing nodes in a heterophilic graph. Surprisingly, the CO variant performs competitively, particularly on heterophilic graphs, suggesting that in some cases, including only the

central node may be sufficient, and that incorporating additional neighbor context can even degrade performance.

These results indicate that for node classification on text-attributed graphs (TAGs), LLMs are often capable of extracting sufficient predictive signals from isolated node semantics, with minimal reliance on explicit structural information. This effectively transforms the graph reasoning task into a set-based problem. We observe a similar trend in link prediction tasks, where structural understanding is typically more critical. Even in this setting, augmenting the input with handcrafted structures such as Laplacian positional encodings provides limited benefit. Instead, **a simple, unordered aggregation of neighboring node descriptions enables the LLM to infer both node semantics and relational connectivity with surprising effectiveness.**

Table 3: This table evaluates whether message passing effectively aggregates useful neighbor information. Comparing a simple MLP baseline with GNN-based adapters, we find that in the LLM setting, message passing can lead to over-smoothing, even with skip connections, reducing the semantic distinctiveness of target nodes. **Best** results are bolded, second best are underlined.

Setting	Dataset	Node Classification			
		MLP	GCN	GAT	GIN
Homophilic	Cora	87.09% (0.66%)	87.64% (0.84%)	**88.25%** (0.53%)	83.03% (5.41%)
	Citeseer	79.39% (1.38%)	**80.20%** (0.13%)	79.74% (0.41%)	79.32% (1.11%)
	Pubmed	**94.76%** (0.10%)	92.24% (1.23%)	92.01% (0.24%)	91.40% (0.63%)
Heterophilic	Shool	**90.17%** (3.62%)	67.87% (3.24%)	64.75% (0.00%)	70.02% (2.19%)
	Roman Empire	**65.39%** (0.29%)	36.51% (18.06%)	36.97% (13.92%)	46.92% (22.37%)
	Amazon Ratings	**40.78%** (0.35%)	40.52% (0.51%)	40.71% (0.23%)	38.76% (0.18%)
Across Datasets		**76.26%**	67.50%	67.07%	68.24%

Setting	Dataset	Link Prediction			
		MLP	GCN	GAT	GIN
Homophilic	Cora	90.72% (0.85%)	90.51% (1.19%)	**91.05%** (0.93%)	87.86% (1.20%)
	Citeseer	87.67% (2.71%)	**89.32%** (0.53%)	88.53% (0.46%)	78.34% (1.99%)
	Pubmed	**89.14%** (0.19%)	89.11% (0.37%)	88.58% (0.38%)	87.54% (0.55%)
Heterophilic	Shool	59.40% (1.92%)	59.40% (3.26%)	**62.78%** (3.98%)	56.55% (1.25%)
	Roman Empire	51.60% (0.62%)	52.64% (0.68%)	51.00% (1.02%)	**53.63%** (0.24%)
	Amazon Ratings	**72.59%** (0.34%)	72.10% (1.04%)	66.24% (11.19%)	71.51% (0.19%)
Across Datasets		75.19%	**75.51%**	74.70%	72.57%

3.3 GNN-Based Encoding

In contrast to LLaGA's template-based structural encoding, several recent studies [Perozzi et al., 2024, He et al., 2024, Wang et al., 2024] have explored the integration of GNN-based modules to inject structural information into LLMs. To further investigate the necessity of such architectural components, we adopt the experimental setup introduced in the previous section and evaluate LLM performance in the absence of explicit structural cues. Our primary focus is on the GraphToken framework [Perozzi et al., 2024], which incorporates GNNs with dynamically constructed graphs during fine-tuning, enabling a flexible and adaptive representation of structural context.

To isolate the contribution of structural modeling, we begin by evaluating the impact of different GNN backbones. Specifically, we replace the GNN with a simple multi-layer perceptron (MLP), while keeping all other components and training configurations constant. This ablation aims to determine whether semantic representations alone can sustain downstream performance without relying on graph-specific inductive biases. As reported in Table 3, although certain GNN architectures may exhibit advantages under specific domain conditions or structural regimes, the overall performance remains largely comparable. This observation aligns with findings from [Perozzi et al., 2024], suggesting that the marginal gains introduced by structural modeling may not justify the added complexity.

Furthermore, we observe that increasing the adapter depth in GraphToken consistently degrades performance when using a GNN module. As shown in Figure 2, deeper GNN-based adapters lead to a significant fluctuation in accuracy, while increasing the number of MLP layer only impacts marginally, indicating potential overfitting or vanishing gains with deeper structural modeling.

Taken together with our earlier observations in the LLaGA setting, these results further challenge the prevailing assumption that structural encoding is critical for LLM-based graph reasoning, **suggesting that for many node classification and link prediction tasks on text-attributed graphs, LLMs can achieve strong performance by leveraging rich semantic signals alone, rendering explicit structural augmentation either redundant or even detrimental in some cases.**

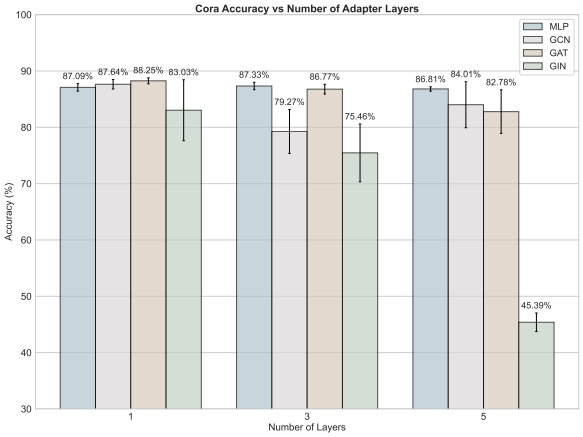


Figure 2: Increasing the number of adapter layers leads to notable performance degradation for GNN-based adapters, particularly GIN, which loses much of its generalizability in deeper configurations. In contrast, MLP adapters, without relying on structural information, maintain stable performance and exhibit greater robustness across varying depths.

4 How Do LLMs Read Natural Graphs?

We have previously demonstrated that structural information can be negligible or even detrimental when it interferes with node-level semantic understanding in the TAG setting. This observation aligns with the intuition that TAG connectivity is often highly correlated with the semantic descriptions of the nodes themselves. As such, LLMs may implicitly reconstruct the graph's connectivity by simply processing the node sequences. However, this raises an important question: *would LLMs behave similarly on graphs that naturally exist, such as molecular structures, where topology is intrinsic rather than semantically induced?*

To investigate this, we conduct experiments on molecular property prediction, a canonical graph-level task. Specifically, we select three datasets from MoleculeNet [Ramsundar et al., 2019]: BACE, BBBP, and HIV, chosen for their diversity in molecular properties and biomedical relevance. Full dataset statistics and preprocessing details are provided in the Appendix.

4.1 Molecular Graphs

Unlike TAG datasets such as citation networks or E-commerce graphs, molecular graphs are typically smaller in scale (fewer nodes) and exhibit lower average node degree, making their topological structures less complex. In such settings, template-based encoding strategies, often used to impose artificial tree-like computational paths, may introduce extraneous structural noise. Therefore, we adopt GNN-based adapters, which are more commonly used for molecular representation learning, to serve as stronger structure-aware baselines.

Interestingly, as shown in Table 4, even a simple MLP head applied to the embeddings of the nodes (atom), without any explicit structural modeling, can perform on par with or even outperform GNN-based adapters. This further supports our hypothesis: LLMs can extract sufficient task-relevant information from node-level semantics alone, rendering explicit structural encoding less critical for downstream performance.

To comprehensively evaluate the role of structural information in LLMs, we conducted experiments across three representative tasks. Across all three, the results consistently suggest that LLMs can operate effectively without leveraging explicit structural information, provided that high-quality node embeddings are available. Notably, the node representations used in our experiments are derived from a pretrained language encoder, ensuring rich semantic content.

Table 4: We further investigate whether structural information provides tangible benefits for LLMs in processing graph-structured data by evaluating three molecular property prediction datasets. Consistent with our earlier findings on TAGs, we observe that plain node embeddings, devoid of any explicit structural encoding, can achieve comparable or even superior performance to structure-aware approaches on molecular tasks.

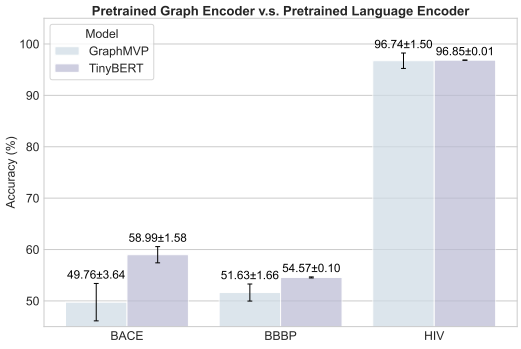
Dataset	Molecular Property Prediction			
	MLP	GCN	GIN	GAT
BACE	**58.99%** (1.66%)	58.77% (9.13%)	**58.99%** (5.52%)	57.46% (3.62%)
BBBP	54.57% (1.38%)	57.84% (0.49%)	**60.29%** (0.49%)	51.96% (1.47%)
HIV	**96.85%** (0.01%)	96.81% (0.03%)	96.79% (0.00%)	96.82% (0.03%)

4.2 Pretrained Graph Encoder v.s. Pretrained Language Encoder

An intriguing follow-up question emerges: what if we replace the language encoder with a pretrained graph encoder? Will structural information, as captured by the graph encoder, play a more central role in enhancing LLM performance?

To further investigate the role of pretrained modality-specific encoders in processing naturally occurring graphs such as molecular structures, we compare embeddings from GraphMVP [Liu et al., 2022], a state-of-the-art graph pretraining framework for molecules, against those from TinyBERT [Jiao et al., 2019], a compact yet effective pretrained language model. For a fair comparison in representation capacity, we match the embedding dimensionalities, using a 5-layer, 300-dimensional GraphMVP and a 4-layer, 312-dimensional TinyBERT. Figure 3 reports the average accuracy along with standard deviations across multiple runs. The results indicate that even in domains where structural priors are intrinsic, such as chemistry, pretrained graph encoders like GraphMVP do not consistently demonstrate a clear advantage in leveraging structural information for LLM-based processing. In contrast, a lightweight pretrained language encoder such as TinyBERT is sufficient to represent molecular graphs solely from sequences of atom-level descriptions, reinforcing our earlier conclusion that LLMs predominantly exploit semantic content rather than explicit structural cues.

Figure 3: How Pretrained Encoders Impact



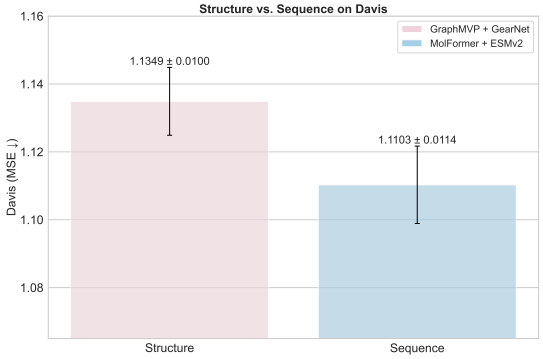
This experiment further reinforces our central finding: LLMs tend to prioritize semantic content over structural information when processing graph-related inputs. Even when structural signals are provided through specialized graph encoders, they fail to surpass the semantic richness embedded within language-based representations. Consequently, our observations point to a broader implication: **the quality and expressiveness of semantic embeddings, rather than explicit graph topology, serve as the dominant factors determining LLM performance on graph-centric reasoning tasks**. This also challenges the conventional assumption that graph-specific pretraining inherently offers a representational advantage in capturing relational and compositional patterns.

4.3 LLMs in Geometric Deep Learning

The current graph benchmarks may not fully capture tasks requiring genuine relational reasoning. In fact, our findings are consistent with Bechler-Speicher et al. [2025], which argues that existing graph benchmarks are limited and often fail to reflect real-world relational complexity. Our results reinforce this position by empirically demonstrating that structural information contributes marginally even in these canonical benchmarks—suggesting that many current datasets may not require, or even reward, structural reasoning.

To further test our hypothesis under more structurally demanding conditions, we extended our experiments to the Davis Drug–Target Interaction (DTI) dataset, a standard benchmark in geometric deep learning (GDL) that explicitly involves molecular–protein structural reasoning. We compared performance between structure-based and sequence-based encoders, we use Llama3-8B as the LLM backbone in this experiment. DTI dataset contains molecules and proteins, and we choose GraphMVP [Liu et al., 2022] and GearNet [Zhang et al., 2023] to encode the molecule and protein structure representations, while for sequence information, we select MolFormer [Ross et al., 2022] and ESMv2 [Lin et al., 2022] as encoders respectively.

Figure 4: Features for LLMs on GDL.



As shown in Figure 4, the sequence-based (semantic) representations perform comparably or even slightly better than structural encodings, reinforcing that our observation generalizes to more challenging, structure-dependent domains. We emphasize that our work and Bechler-Speicher et al. [2025] share a similar perspective: the field should move toward rethinking graph benchmarks and embracing semantics- or geometry-aware tasks as complementary directions for future graph learning research.

5 What affects LLMs in understanding Graphs?

While our experiments suggest that LLMs may not inherently benefit from explicit structural information, it is important to recognize that their ability to leverage such signals can vary significantly depending on factors such as pretraining corpus, optimization strategy, and model scale. To assess the robustness of our findings, we further investigate whether the observed trends hold consistently across different backbone LLM architectures and parameter sizes. This analysis aims to disentangle model-specific artifacts from generalizable behavior, and to evaluate whether the limited utility of structural encodings persists regardless of underlying model configurations.

5.1 Scaling Ineffectiveness

It is widely acknowledged that increasing the parameter size of LLMs often leads to enhanced expressive power and improved performance across a broad range of tasks. To assess whether this scaling trend extends to graph-related tasks, we evaluate the impact of model size on the ability of LLMs to utilize structural information. Specifically, in Table 5, we compare the structure-aware ND template with the structure-free HN template using two model variants: LLaMA2-7B and LLaMA2-13B [Touvron et al., 2023].

Our results reinforce the patterns observed in previous experiments. Despite increasing the backbone model size, the tendency of LLMs to overlook explicit structural encodings remains consistent. Notably, scaling up to 13B parameters does not enhance the model's ability to leverage structural information. In fact, in most cases, the structure-free HN template outperforms the structure-aware ND template, further suggesting that model scale alone does not improve sensitivity to structural signals in graph-based tasks.

5.2 Semantic Content

To further assess the robustness of our findings, we investigate whether the reliance on structural information changes under weaker semantic content. Specifically, we reduce the descriptive richness of each node by comparing two settings: (1) full node descriptions, such as full abstracts or complete webpage content, and (2) sparse descriptions, limited to titles of papers or webpages. We generate node embeddings using three widely used pretrained models: RoBERTa-large [Liu et al., 2019], BERT-large [Devlin et al., 2018], and T5-XXL [Raffel et al., 2020].

Even under reduced semantic conditions, the structure-free HN template consistently matches or outperforms the structure-aware ND template (see Table 5). These results suggest that LLMs are

Table 5: Switching LLM backbones preserves our finding that structure may be unnecessary for LLMs processing graphs. Even with weak semantic content, LLMs still reveal the same pattern.

Model Architecture	Dataset	Node Classification		Link Prediction	
		ND	HN-1	ND	HN-1
Llama2-7B	Cora	87.76%(0.21%)	**88.01%**(0.56%)	85.48%(0.38%)	**87.04%**(0.75%)
	School	70.98%(0.83%)	**92.09%**(2.49%)	61.82%(2.88%)	**69.09%**(1.92%)
Llama2-13B	Cora	**87.58%**(0.59%)	87.45%(0.19%)	84.24%(0.89%)	**86.05%**(0.55%)
	School	69.30%(3.24%)	**89.45%**(3.40%)	61.21%(1.28%)	**67.15%**(1.52%)

Semantic Content	Dataset	Node Classification		Link Prediction	
		ND	HN-1	ND	HN-1
sparse	Cora	**83.96%**(2.74%)	82.17%(0.56%)	69.19%(1.15%)	**74.81%**(0.85%)
	School	56.95%(6.19%)	**73.62%**(7.21%)	63.63%(0.63%)	**65.09%**(3.93%)
full	Cora	83.39%(0.37%)	**84.81%**(0.46%)	70.81%(1.89%)	**75.84%**(0.74%)
	School	59.47%(3.97%)	**60.19%**(1.10%)	63.15%(5.91%)	**70.06%**(3.30%)

capable of extracting meaningful relational patterns from minimal semantic cues, without the need for explicit structural encodings. This further reinforces our conclusion that structural augmentation provides limited benefits, even when node-level semantics are sparse.

5.3 Will Dataset Size Impact This Finding?

Table 6: Our findings hold consistently on larger text-attributed graphs, suggesting that structural information contributes only marginally to LLMs' graph inference.

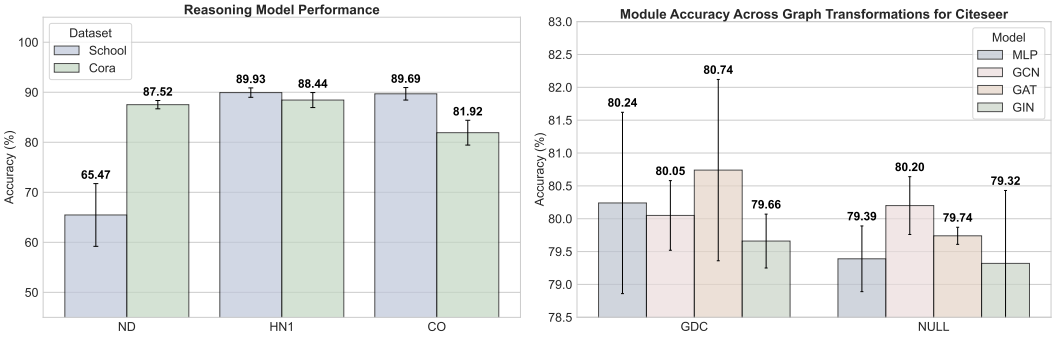
Dataset	Node Classification		
	ND	HN-1	CO
Products	83.45% (0.39%)	**83.87%** (0.24%)	80.10% (0.27%)
ArXiv	**75.65%** (0.50%)	75.41% (0.21%)	74.46% (0.18%)

To assess whether structural information benefits LLMs on larger graph datasets, we compared node classification performance under two settings: ND (structure-aware) and HN-1 (structure-agnostic). Experiments were conducted on the PRODUCTS [Bhatia et al., 2016] and ARXIV [Wang et al., 2020] benchmarks (Table 6), with the null hypothesis that both settings yield comparable mean performance. Each configuration was evaluated over three runs. The results show no statistically significant difference between ND and HN-1 on either dataset (two-sample Welch's t-test: $p_{\text{products}} = 0.20 > 0.05$, $p_{\text{arxiv}} = 0.49 > 0.05$), indicating that structural information does not produce consistent gains. This finding aligns with our earlier results, suggesting that LLMs effectively capture relational dependencies without relying on explicit graph topology.

5.4 How about Large Reasoning Models?

Large Reasoning Models (LRMs) aim to replicate human-like problem solving by drawing conclusions from structured rules and evidence. In this context, template-based graph encodings can be viewed as implicit reasoning prompts that guide the model's attention. This motivates us to explore whether explicit structural signals, like Laplacian positional encodings, can enhance reasoning performance. We conduct a preliminary evaluation using OpenReasoning-Nemotron-7B [Ahmad et al., 2025], a large-scale model post-trained for reasoning in math, science, and code domains on Cora and School datasets.

Figure 5: *Left*: Though reasoning model can perform structured decision-making, it does not rely on structure information. *Right*: Altering the node sequence via GDC can gain some enhancement at a time.



Consistent with our earlier observations, we find that LRMs, despite their specialized training for structured reasoning, **do not exhibit significant improvements when structural encodings such as Laplacian embeddings are included**. These results further reinforce our central finding: even in models explicitly designed for reasoning, the addition of graph structural signals does not necessarily translate to better generalization or task performance. This suggests that the current generation of reasoning models primarily rely on semantic representations and may underutilize explicit graph structure unless such reasoning is explicitly aligned with the model's pretraining or task formulation.

6 Can LLMs Better Leverage Structure?

Although structural information appears to have limited influence on LLM performance, we observed that the GraphToken framework, when paired with an MLP adapter, occasionally outperforms the structure-free CO template. While MLPs lack explicit message-passing mechanisms, they may still benefit from implicit structural cues preserved in the ordering of the input node sequence. Inspired by this observation, we hypothesize that optimizing node sequence selection can better expose latent structural signals to the LLM and potentially improve the performance.

To investigate this, we did a preliminary study to incorporate Graph Diffusion Convolution (GDC) [Gasteiger et al., 2019], a graph transformation technique designed to capture long-range dependencies using a sparsified generalized form of graph diffusion. GCD implicitly generates a new graph by graph diffusion as well as a following sparsification step, so the information can be aggregated from a larger neighborhood. As a result, applying GDC can also effectively condenses the input node sequence into a sparse, center-focused subset, resulting in improved LLM performance in most cases (as illustrated in Figure 5). While this does not overturn our main finding, it highlights a promising direction: certain graph transformations, especially those capturing longer-range information, may provide a structured yet minimal signal that LLMs can exploit more effectively. Future work may further investigate how to integrate such transformations with semantic guidance to better align graph structure with LLM capabilities.

7 Conclusion and Future Directions

In this study, we revisited LLM-based approaches to TAG tasks and systematically evaluated different structural encoding strategies. We find that LLMs largely treat graphs as unordered sets, showing minimal sensitivity to explicit structural cues from input templates or model-level components such as GNNs. These findings challenge the conventional view that structural information is essential for graph reasoning, highlighting instead the dominant role of semantics in LLM-based graph learning. **Our results provide an empirical foundation for understanding LLM–graph interactions and underscore the importance of effective node sequencing over structural encodings in advancing LLM performance on TAG tasks.**

References

Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. *Advances in Neural Information Processing Systems*, 35:22326–22340, 2022. 1

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163, 2000. 1, 3

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, Sep. 2008a. doi: 10.1609/aimag.v29i3.2157. URL https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2157. 1

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 1, 2

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ. accepted as poster. 1, 2

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km. 1

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 1, 2

Jie Chen, Tengfei Ma, and Cao Xiao. FastGCN: Fast learning with graph convolutional networks via importance sampling. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rytstxWAW. 1

Haotao Wang, Ziyu Jiang, Yuning You, Yan Han, Gaowen Liu, Jayanth Srinivasa, Ramana Rao Kompella, and Zhangyang Wang. Graph mixture of experts: Learning on large-scale graphs with explicit diversity modeling. In *NeurIPS*, 2023. 1

Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. Attending to graph transformers. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=HhbqHBBrfZ. 1

Kai Neubauer, Yannick Rudolph, and Ulf Brefeld. Toward principled transformers for knowledge tracing, 2024. URL https://openreview.net/forum?id=4dtwyV7XyW. 1

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=OeWooOxFwDa. 1

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. LLaGA: Large language and graph assistant. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 7809–7823. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chen24bh.html. 1, 3, 4

Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. LLMs as zero-shot graph learners: Alignment of GNN representations with LLM token embeddings. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=32g9BWTndc. 1, 3, 5

Bryan Perozzi, Bahare Fatemi, Dustin Zelle, Anton Tsitsulin, Mehran Kazemi, Rami Al-Rfou, and Jonathan Halcrow. Let your graph do the talking: Encoding structured data for llms, 2024. URL https://arxiv.org/abs/2402.05862. 1, 3, 4, 5

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. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=MPJ3oXtTZl. 1, 3, 5

Maya Bechler-Speicher, Ido Amos, Ran Gilad-Bachrach, and Amir Globerson. Graph neural networks use graphs when they shouldn't. In *Forty-first International Conference on Machine Learning*, 2024. 1

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017. 2

Maya Bechler-Speicher, Ben Finkelshtein, Fabrizio Frasca, Luis Müller, Jan Tönshoff, Antoine Siraudin, Viktor Zaverkin, Michael M Bronstein, Mathias Niepert, Bryan Perozzi, et al. Position: Graph learning will lose relevance due to poor benchmarks. *arXiv preprint arXiv:2502.14546*, 2025. 2, 7, 8

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 2

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5812–5823. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/3fe230348e9a12c13120749e3f9fa4cd-Paper.pdf. 2

Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. Pre-training of graph augmented transformers for medication recommendation. *arXiv preprint arXiv:1906.00346*, 2019. 2

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020. 2

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 (WWW'23)*, 2023. 2

Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural network for semi-supervised learning on graphs. In *NeurIPS'20*, 2020. 2

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xQUe1pOKPam. 2, 7, 8

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models, 2024. 3

Xi Zhu, Haochen Xue, Ziwei Zhao, Wujiang Xu, Jingyuan Huang, Minghao Guo, Qifan Wang, Kaixiong Zhou, and Yongfeng Zhang. Llm as gnn: Graph vocabulary learning for text-attributed graph foundation models. *arXiv preprint arXiv:2503.03313*, 2025. 3

Yuyao Ge, Shenghua Liu, Baolong Bi, Yiwei Wang, Lingrui Mei, Wenjie Feng, Lizhe Chen, and Xueqi Cheng. Can graph descriptive order affect solving graph problems with LLMs? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6404–6420, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.321/. 3

Haoyu Wang, Shikun Liu, Rongzhe Wei, and Pan Li. Model generalization on text attribute graphs: Principles with large language models. *arXiv preprint arXiv:2502.11836*, 2025. 3

Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. Can LLMs effectively leverage graph structural information through prompts, and why? *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=L2jRavXRxs. 3

Xixi Wu, Yifei Shen, Fangzhou Ge, Caihua Shan, Yizhu Jiao, Xiangguo Sun, and Hong Cheng. When do llms help with node classification? a comprehensive analysis. In *International Conference on Machine Learning*. PMLR, 2025. URL https://arxiv.org/abs/2502.00829. 3

Chuang Zhou, Zhu Wang, Shengyuan Chen, Jiahe Du, Qiyuan Zheng, Zhaozhuo Xu, and Xiao Huang. Taming language models for text-attributed graph learning with decoupled aggregation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers), pages 3463–3474, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.173/. 3

Yukun Cao, Shuo Han, Zengyi Gao, Zezhong Ding, Xike Xie, and S Kevin Zhou. GraphInsight: Unlocking insights in large language models for graph structure understanding. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12096–12134, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.591/. 3

Ziyin Zhang, Hang Yu, Sage Lee, Peng Di, Jianguo Li, and Rui Wang. GALLa: Graph aligned large language models for improved source code understanding. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13784–13802, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.676/. 3

Zhong Guan, Likang Wu, Hongke Zhao, Ming He, and Jianpin Fan. Attention mechanisms perspective: Exploring llm processing of graph-structured data. *arXiv preprint arXiv:2505.02130*, 2025. 3

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*, pages 89–98, 1998. 3

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

Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Seán Slattery. Learning to extract symbolic knowledge from the world wide web. *AAAI/IAAI*, 3 (3.6):2, 1998. 3

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? *arXiv preprint arXiv:2302.11640*, 2023. 3

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 4, 15

Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019. https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837. 6

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019. 7

Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *International Conference on Learning Representations*, 2023. 8

Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022. doi: 10.1038/s42256-022-00580-7. 8

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022. 8

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 8

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692. 8

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805. 8

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html. 8

K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016. URL http://manikvarma.org/downloads/XC/XMLRepository.html. 9

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1): 396–413, 2020. 9

Wasi Uddin Ahmad, Sean Narenthiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. OpenCodeReasoning: Advancing Data Distillation for Competitive Coding. 2025. URL https://arxiv.org/abs/2504.01943. 9

Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. *Advances in neural information processing systems*, 32, 2019. 10

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6. 15

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 15

A Dataset Details

In this section, we will introduce our used datasets in details:

- Cora: The Cora dataset is a classic citation network where each node represents a machine learning research paper, and edges indicate citation relationships between papers. Each paper is described by a sparse bag-of-words feature vector, and the task is to classify papers into one of seven predefined categories such as neural networks or case-based reasoning. Total 2,708 nodes will be classified into {'Theory', 'Neural Networks', 'Probabilistic Methods', 'Reinforcement Learning', 'Case Based', 'Rule Learning', 'Genetic Algorithms'}
- Citeseer: Citeseer is another widely-used citation network dataset in which nodes represent research papers and edges denote citation links. Each node includes word-based features and belongs to one of six scientific categories. These labels {'artificial intelligence', 'human-computer interaction', 'information retrieval', 'database', 'agents', 'machine learning'} will be associated to 3,186 nodes in Citeseer.
- Pubmed: The Pubmed dataset is a large-scale citation graph composed of scientific papers from the biomedical domain. Each node represents a paper described by a TF/IDF-weighted word vector from the paper's abstract, and edges correspond to citation links. Pubmed contains 19,717 nodes, and nodes are partitioned into 3 label categories: {Diabetes Mellitus Type1, Diabetes Mellitus Type2, Diabetes Mellitus Experimental}
- School: School dataset is a collection of 4 common heterophilic graph datasets: Cornell, Texas, Washington, and Wisconsin. All of these 4 datasets are from the WebKB collection, where represent web pages from {Cornell University, University of Texas, University of Washington, University of Wisconsin} correspondingly and edges capture hyperlinks between them. Model needs to classify each node (webpage) into 5 categories: 'project', 'course', 'student', 'faculty', 'staff', and 'student'. The total number of nodes in School dataset is 872.
- Roman Empire: Roman Empire dataset is a synthetic temporal graph dataset designed to evaluate temporal graph learning models. There are 17 labels in total: {'passive subject', 'coordinating conjunction', 'active subject', 'object of preposition', 'adverbial modifier', 'adjective modifier', 'relative clause', 'noun compound modifier', 'appositive modifier', 'prepositional marker',

'passive auxiliary', 'possessive modifier', 'direct object', 'null', 'conjoined element', 'auxiliary verb', 'main predicate', 'determiner'}, and Roman Empire contains 24,492 nodes.

- Amazon Ratings: The Amazon Ratings dataset represents a temporal bipartite graph where nodes are users and products, and edges correspond to product ratings over time. There are 24,492 comments with 5 different rating scales: {'excellent – exceeded all expectations', 'very good – almost perfect, just shy of excellent', 'decent – some good, some bad', 'good – solid experience with minor flaws', 'terrible – extremely disappointing'}
- BACE: contains bioactivity data for small molecules that inhibit human β-secretase 1 (BACE-1), a key target in Alzheimer's disease drug discovery. Each molecule is labeled as active or inactive, making it a binary classification task for molecular binding affinity.
- BBBP: consists of compounds labeled according to their ability to penetrate the blood-brain barrier, which is crucial in central nervous system drug design. This is also a binary classification task.
- HIV: includes information on over 40,000 compounds tested for their ability to inhibit HIV replication. Each molecule is labeled as active or inactive against HIV, making it another binary classification problem aimed at identifying potential antiretroviral candidates.

Each dataset follow the same train-test split ratio 8:2.

B Experiment Configuration

dataset	training epoch	total training time
Cora	5	$\sim$ 16mins
Citeseer	5	$\sim$ 10mins
Pubmed	1	$\sim$ 9mins
School	13	$\sim$ 3mins
Roman Empire	1	$\sim$ 10mins
Amazon Ratings	1	$\sim$ 10mins
BACE	12	$\sim$ 5mins
BBBP	8	$\sim$ 5mins
HIV	3	$\sim$ 8mins

Table 7: Configuration and efficiency estimation for each dataset.

Each dataset is trained on 8 A6000 GPUs, and the training batch size is set to 4 per GPU for all dataset, and the learning rate for template-based encoding is 2e-3 and for GNN-based encoding is 1e-4. We use AdamW optimizer and DeepSpeed to perform the multi-GPU training. We use the vicuna-7b [Zheng et al., 2023] as our main LLM backbone for all experiments. We report average results from 3 random seed runs. For GraphToken experiments, we set the number of adapter layer at 1 for each adapter module. Setting adapter layer at 1 usually offers the best performance, and model will easily lose its expressivity with a deeper adapter layer. All models and experiments are built using Hugging Face [Wolf et al., 2020] and torch geometric [Fey and Lenssen, 2019] packages.

C Prompts

- Cora: Given a node-centered graph: $<$ graph $>$, each node represents a paper, we need to classify the center node into 7 classes: Case Based, Genetic Algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning, Theory, please tell me which class the center node belongs to?
- Citeseer: Given a node-centered graph: $<$ graph $>$, each node represents a paper, we need to classify the center node into 6 classes: Agents, Machine Learning, Information Retrieval, Database, Human-Computer Interaction, Artificial Intelligence, please tell me which class the center node belongs to?
- Pubmed: Given a node-centered graph: $<$ graph $>$, each node represents a paper about Diabetes, we need to classify the center node into 3 classes: Diabetes Mellitus Experimental, Diabetes Mellitus Type1, Diabetes Mellitus Type2, please tell me which class the center node belongs to?

- School: In a graph of a university website, each node represents a web page, and each edge indicates that one web page links to another via a hyperlink. The web pages can belong to one of the following categories: project, faculty, course, student, staff. Here is a node-centered graph: $<$ graph $>$, what is the category?
- Roman Empire: In an article, words that have dependency relationships (where one word depends on another) are connected, forming a dependency graph. Based on the connections between words, determine the syntactic role of each word. Given that a word described in a node-centered graph: $<$ graph $>$, what is this word syntactic role?
- Amazon Ratings: n a product graph dataset, edges connect products that are frequently purchased together. Based on the connections between products (books, music CDs, DVDs, VHS tapes), predict the average rating given by reviewers for the products. Given that a product described in a node-centered graph: $<$ graph $>$, what is the product rating?
- BACE: Given the following molecule $<$ graph $>$, determine whether it is active or inactive as a BACE-1 inhibitor.
- BBBP: Determine whether the following molecule $<$ graph $>$ can penetrate the blood-brain barrier (BBB) based on its SMILES representation.
- HIV: This molecule is represented by the following $<$ graph $>$. Predict whether it is active or inactive against HIV replication.

The $<$ graph $>$ serves as a placeholder token, which will be replace by the input node sequence during training and inference stages.