LLMs Do Not Read Text-Attributed Graphs as Effectively as We Expected

Anonymous ACL submission

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**. These findings challenge the necessity of explicit graph structure in the LLM era and suggest a need to rethink graph learning paradigms in light of powerful language models.

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 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 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. These findings challenge 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.

2 Related Work

Graph Learning: Graph learning offers a robust framework for modeling relational data across do-

mains like social networks, biology, and knowledge graphs. At its core, Graph Neural Networks (GNNs) learn node and graph representations through message passing and aggregation (Kipf and Welling, 2017; Hamilton et al., 2017), with variants like Graph Attention Networks (Veličković et al., 2018) and spectral methods (Bruna et al., 2013) enhancing scalability and expressiveness. Inspired by NLP and vision, self-supervised methods such as GraphCL (You et al., 2020), G-BERT (Shang et al., 2019), and GPT-GNN (Hu et al., 2020) use contrastive or masked objectives to improve generalization. However, the lack of standardized benchmarks and input formats hampers cross-domain transfer. In response, Graph Foundation Models (GFMs) like GraphMAE (Zhenyu Hou, 2023), GRAND (Feng et al., 2020), and GraphMVP (Liu et al., 2022) aim to learn general-purpose representations, though issues like data heterogeneity and vocabulary gaps persist, fueling interest in leveraging LLMs for more scalable graph representation learning.

LLMs as GFMs: Recent advances move beyond traditional GNN-based Graph Foundation Models (GFMs) by positioning LLMs as graph learners, leveraging their generalization and multimodal reasoning abilities. Studies such as Fatemi et al. (2024) show that LLM performance on graph tasks is sensitive to graph-to-text encoding strategies, task types, and structural priors. OFA (Liu et al., 2023) introduces a unified task formulation using natural language prompts around nodes-of-interest, while LLaGA (Chen et al., 2024) refines this idea by applying structure-aware node reordering to better align graph inputs with LLM processing. PromptGFM (Zhu et al., 2025) adds an in-text graph vocabulary to unify LLMs and GNNs, and LLM-BP (Wang et al., 2025) enhances reasoning by combining belief propagation with LLM-inferred homophily.

Hybrid models like GraphToken (Perozzi et al., 2024) use GNN adapters and prompts to inject structure into LLMs, while G-Retriever (He et al., 2024) and TEA-GLM (Wang et al., 2024) further fuse structural and semantic cues for strong benchmark performance. However, recent findings by Guan et al. (2025) reveal that transformer attention often fails to reflect true graph topology, suggesting that limitations lie in LLMs' internal processing of structure, rather than their downstream potential.

3 Do LLMs Read TAG as Expectation?

In graph learning, models typically fuse semantic node features with structural connectivity, a core principle behind the success of GNNs. Inspired by this, recent work has explored injecting structural signals into LLMs, either through handcrafted templates, as in LLaGA (Chen et al., 2024), or via GNN-based adapters like GraphToken, G-Retriever, and TEA-GLM (Perozzi et al., 2024; He et al., 2024; Wang et al., 2024), which learn structural embeddings.

These methods fall into two categories: template-based approaches that manually encode neighborhoods, and GNN-based ones that learn structure through neural encoders. However, our findings reveal that both offer similar and often marginal gains when node semantics are strong, indicating that LLMs primarily rely on content rather than topology. This challenges the necessity of structural encoding in text-rich graphs and suggests a shift toward more minimalist, semantics-driven graph foundation models.

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

3.1 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 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?

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.

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		
		ND	HN-1	CO
Homophilic	Cora	88.07% (0.74%)	**88.56%** (0.80%)	85.42% (1.78%)
	Citeseer	**80.31%** (0.81%)	80.20% (0.94%)	77.74% (0.31%)
	Pubmed	92.56% (0.71%)	94.80% (0.17%)	**94.84%** (0.04%)
Heterophilic	Shool	66.43% (3.69%)	82.02% (12.79%)	**91.13%** (1.66%)
	Roman Empire	48.56% (1.17%)	59.70% (2.42%)	**62.24%** (0.19%)
	Amazon Ratings	40.97% (0.56%)	**41.67%** (0.22%)	40.38% (1.14%)
Across Datasets		69.48%	74.49%	**75.29%**

3.2 Template-Based Encoding

In this subsection, we revisit the LLaGA framework (Chen et al., 2024), with a particular focus on the *Neighborhood Detail (ND)* template. This template relies on a predefined computational graph, typically a k-hop B-tree, and uses Laplacian-based positional encodings to inject structural signals into the LLM input. To assess the utility of these structural components, we perform a systematic ablation by removing both the handcrafted subgraphs and positional encodings, replacing them with a simple, order-agnostic sequence of node descriptions.

We compare the original ND template against two lightweight, structure-agnostic baselines: (1) **HN (Hop Neighbor)**, which randomly samples a subset of k-hop neighbors to form a node sequence, and (2) **CO (Center Only)**, which includes only the description of the central node. As shown in Table 2, the ND template does not significantly outperform either baseline. Compared to the non-Laplacian HN variant, which uses plain neighbors without positional encodings, we observe more robust and consistent performance across both homophilic and heterophilic graphs. Surprisingly, even the CO baseline achieves competitive results, especially on heterophilic graphs, suggesting that including only the center node may suffice, and that additional neighbor information can sometimes hinder the model's understanding.

These findings indicate that, for node classification tasks on TAGs, LLMs can extract sufficient predictive signals from individual node semantics, with limited benefit from explicit structures, transforming the graph problem into a set problem.

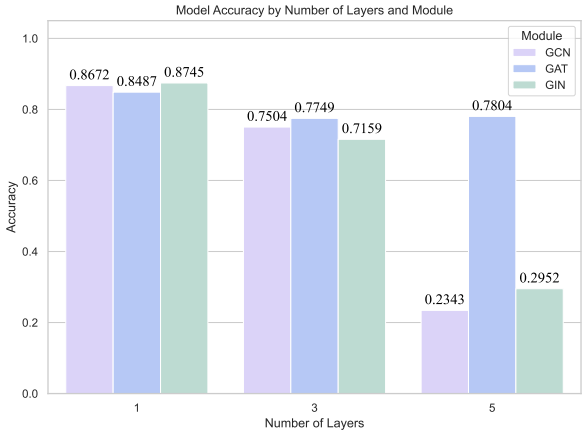


Figure 1: Having deeper adapter will easily make model lose it generalization ability for GraphToken approach.

3.3 GNN-Based Encoding

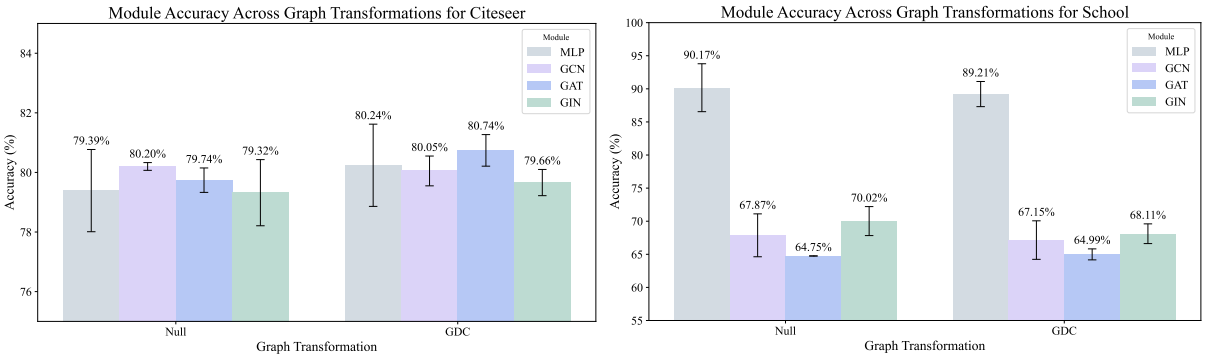
In contrast to LLaGA's template-based approach to structural encoding, several recent studies (Perozzi et al., 2024; He et al., 2024; Wang et al., 2024) have explored integrating GNN-based modules to inject structural information into LLMs. Following the experimental setup from the previous section, we examine how LLMs perform in the absence of such structural cues. Our primary focus is on the GraphToken framework (Perozzi et al., 2024), which trains GNNs with different dynamic structures during fine-tuning, enabling flexible and adaptive input graph.

We begin by evaluating the impact of different GNN architectures. As shown in Table 3, while some GNNs may be better suited to specific text domains or graph structures, overall performance remains comparable, consistent with findings in (Perozzi et al., 2024). To isolate the role of structural modeling, we replace the GNN with a simple

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.

Setting	Template	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%

Figure 2: Altering the node sequence via GDC can gain some enhancement at a time.



multi-layer perceptron (MLP), keeping all other components and training settings fixed. This ablation helps determine whether semantic features alone can support downstream performance without graph-specific inductive biases. Additionally, we observe that GraphToken's performance degrades with increased adapter depth when using a GNN module. As shown in Figure 1, adding more layers leads to a consistent drop in accuracy.

Consistent with our earlier findings from LLaGA, LLMs achieve competitive performance even without GNN modules. This challenges the common assumption that structural encodings are essential, suggesting instead that textual semantics alone may suffice for many node classification tasks in textual attribute graphs.

4 Can LLMs Better Leverage Structure?

Although structural information appears to have limited impact on LLM performance, GraphToken with an MLP adapter occasionally outperforms the CO template. While MLPs lack explicit message passing, they may still benefit from implicit structural cues embedded in the node sequence. This highlights a promising direction: systematically selecting node sequences to better exploit structural signals.

To explore this, we apply Graph Diffusion Convolution (GDC) (Gasteiger et al., 2019), a graph transformation that captures long-range dependencies. As shown in Figure 2, GDC can condense the node sequence into a center-focused subset and improve LLM performance, suggesting that structure-aware sparsification can be beneficial. Since GDC uses only graph connectivity and ignores node semantics, future work could integrate both structural and semantic signals to guide graph transformations tailored for LLMs.

5 Conclusion and Future Directions

In this study, we revisit LLM-based methods for node classification on TAGs. Our findings reveal that LLMs largely treat graphs as unordered sets, regardless of structural information added at the input or model level. Accurate predictions can be made using only the center node and the node sequence, indicating limited influence of structural signals. This underscores the importance of designing effective node sequences to better harness LLMs for TAG tasks.

Limitations

In our work, we focus on assessing the importance of structural information in Text-Attributed Graphs, which are inherently 2D and primarily capture topological relationships. However, structural information can extend beyond topology to include geometric properties such as 3D coordinates, which often play a more critical role in downstream performance. Future research could explore domains like molecular and protein structures, where geometric information is both natural and essential for accurate modeling.

Acknowledgments

We acknowledge the use of AI tool (ChatGPT/GPT 4o) (Hurst et al., 2024) for paraphrasing or polishing our original content. After polishing by AI, we also check the correctness to make sure there is not hallucinated sentences. We also provide sufficient citation in the Reference section. Our idea and experiment design are not relied on AI tools.

References

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

Maya Bechler-Speicher, Ben Finkelshtein, Fabrizio Frasca, Luis Müller, Jan Tönshoff, Antoine Siraudin, Viktor Zaverkin, Michael M Bronstein, Mathias Niepert, Bryan Perozzi, and 1 others. 2025. Position: Graph learning will lose relevance due to poor benchmarks. *arXiv preprint arXiv:2502.14546*.

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

Jie Chen, Tengfei Ma, and Cao Xiao. 2018. FastGCN: Fast learning with graph convolutional networks via importance sampling. In *International Conference on Learning Representations*.

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

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

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

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

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

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

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

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

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

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NIPS*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2023. One for all: Towards training one graph model for all classification tasks. *arXiv preprint arXiv:2310.00149*.

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. 2022. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations*.

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

Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. 2024. Attending to graph transformers. *Transactions on Machine Learning Research*.

Kai Neubauer, Yannick Rudolph, and Ulf Brefeld. 2024. Toward principled transformers for knowledge tracing.

Bryan Perozzi, Bahare Fatemi, Dustin Zelle, Anton Tsitsulin, Mehran Kazemi, Rami Al-Rfou, and Jonathan Halcrow. 2024. Let your graph do the talking: Encoding structured data for llms. *Preprint*, arXiv:2402.05862.

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

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

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

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

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*. Accepted as poster.

Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. 2024. 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*.

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

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

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, and 3 others. 2020. 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. Association for Computational Linguistics.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? In *International Conference on Learning Representations*.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, volume 33, pages 5812–5823. Curran Associates, Inc.

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

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. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Yukuo Cen Xiao Liu Yuxiao Dong Evgeny Kharlamov Jie Tang Zhenyu Hou, Yufei He. 2023. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proceedings of the ACM Web Conference 2023 (WWW'23)*.

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

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'}

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

Table 4: 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?

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