GaLoRA: Parameter-Efficient Graph-Aware LLMs for Node Classification

Mayur Choudhary **Saptarshi Sengupta** **Katerina Potika**
Department of Computer Science, San Jose State University
{mayur.choudhary, saptarshi.sengupta, katerina.potika}@sjsu.edu

Abstract

The rapid rise of large language models (LLMs) and their ability to capture semantic relationships has led to their adoption in a wide range of applications. Text-attributed graphs (TAGs) are a notable example where LLMs can be combined with Graph Neural Networks (GNNs) to improve the performance of node classification. In TAGs, each node is associated with textual content, and such graphs are commonly seen in various domains such as social networks, citation graphs, recommendation systems, etc. Effectively learning from TAGs would enable better representations of both structural and textual representations of the graph and improve decision-making in relevant domains. We present GaLoRA, a parameter-efficient framework that integrates structural information into LLMs. GaLoRA demonstrates competitive performance on node classification tasks with TAGs, performing on par with state-of-the-art models with just 0.24% of the parameter count required by full LLM fine-tuning. We experiment with three real-world datasets to showcase GaLoRA's effectiveness in combining structural and semantical information on TAGs.

1 Introduction

TAGs present a unique challenge in machine learning by requiring the simultaneous representation of both graph structure and rich textual information [1, 2]. Traditional approaches typically leverage GNNs to capture structural dependencies [3] or Pretrained Language Models (PLMs) to process semantic content [4]. Joint models that combine both have shown promise but are often computationally expensive and difficult to scale.

We propose GaLoRA (Graph-aware Low-Rank Adaptation), a modular and efficient framework that enables LLMs to incorporate structural information from graphs without requiring joint training. GaLoRA follows a two-phase design: the first in which a GNN learns structure-aware embeddings that are later injected into the LLM during fine-tuning using Low-Rank Adaptation [5]. This decoupling reduces training overhead while preserving the benefits of structure-semantic fusion.

Through experiments on various TAG datasets, we demonstrate that GaLoRA achieves performance on par with recent baselines [6] while requiring significantly fewer parameters, making it ideal for deployment in resource-constrained real-world settings.

2 Related Work

Recent approaches have explored combining GNNs and PLMs to address the challenges of learning on text-attributed graphs. One such method is GLEM [7], which uses the Expectation-Maximization (EM) framework to train the GNN and LLM in alternating steps. GLEM uses pseudo-labels to iteratively refine both models without requiring joint backpropagation, thereby reducing computational

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: NPGML.

cost. However, its reliance on pseudo-label quality makes it sensitive to noise, and its iterative nature may still be computationally heavy for large-scale graphs.

Another efficient integration method is TAPE [8], which leverages LLMs as a service to generate explanations and predictions via prompting, and then uses a smaller language model to convert these explanations into embeddings consumable for GNNs. This modular approach avoids fine-tuning the LLM, significantly reducing computation. However, its strong dependence on manually crafted prompts and the variable interpretability of LLM-generated explanations can lead to inconsistent performance.

More recently, GraphAdapter [6] introduced a parameter-efficient strategy that freezes the LLM and incorporates a lightweight GNN adapter to inject graph structure into the LLM's hidden representations. While this approach offers scalability and generalization, it may limit the model's ability to adapt to task-specific semantic knowledge, as the LLM remains frozen throughout. In contrast to the discussed works, our proposed framework, GaLoRA, aims to preserve modularity and efficiency by decoupling the training of GNN and LLM modules, while still allowing direct integration of structural information into the LLM through LoRA-based adaptation during fine-tuning.

3 Methodology

To provide a modular approach that is efficient, we present GaLoRA, a framework to fine-tune LLMs with graph context using parameter-efficient training. The framework decouples structural and semantic learning; it trains the GNN and LLM modules separately and integrates the learned representation by injecting structural embeddings into the language model during fine-tuning. With the use of this framework, the goal is to fine-tune the language model such that it aligns with both the textual content and the structural information of the TAG. To ensure the language model adaptation is efficient, the framework utilises LoRA [5] by adding small trainable low-rank matrices into the frozen layers of the pretrained language model. This enables the LLM to efficiently train and achieve the small delta required for the task-specific adaptation.

The framework operates in two distinct phases. In the first phase, a GNN model is trained on the TAG for the node classification task to extract rich structure-aware node embeddings. In the second phase, the LLM is fine-tuned on a separate node classification task using the node text embeddings as the input. During the fine-tuning, the structure-aware embeddings are injected into selected layers of LLMs, enabling them to incorporate the structural information along with the semantic understanding of the attached text. This modular design decouples structure and language learning while still allowing the LLM to benefit from the structural context.

3.1 Phase-1: GNN Training

The first component of the framework is a GNN that models the structural dependencies between nodes. Each node is represented by a feature vector derived from the textual content associated with it. These initial node embeddings are obtained using a LLM encoder, identical to the language model employed in the second phase of the framework. In our experiments, we primarily use GraphSAGE [3] as the GNN, due to its effectiveness in aggregating information from neighboring nodes.

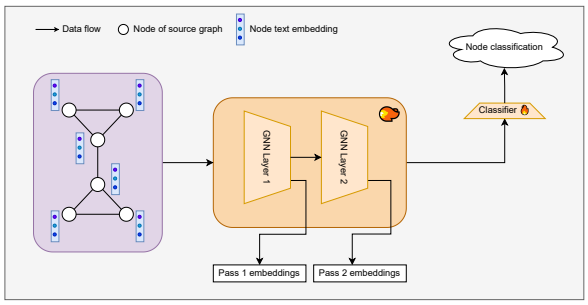

Figure 1: GaLoRA Phase-1 training

Figure 1 shows that the architecture has two message passing layers present in the GNN model, which help capture the 1-hop and 2-hop neighborhood for each node. The first layer output is stored in an intermediate matrix called Pass-1, and the second layer output is similarly stored in the Pass-2 matrix. The information aggregation from the neighbors is done using mean pooling, which is then followed by a non-linear transformation.

Pass-1 (1-hop aggregation):

$$H_v^{(1)} = \text{ReLU}\left(W^{(0)} \cdot \text{CONCAT}\Big(X_v,\ \text{MEAN}\{X_u : u \in \mathcal{N}(v)\}\Big) + b^{(0)}\right) \tag{1}$$

Pass-2 (2-hop aggregation):

$$H_v^{(2)} = \text{ReLU}\left(W^{(1)} \cdot \text{CONCAT}\Big(H_v^{(1)},\ \text{MEAN}\{H_u^{(1)} : u \in \mathcal{N}(v)\}\Big) + b^{(1)}\right) \tag{2}$$

where:

- $X_v \in \mathbb{R}^d$ is the text embedding of node v,
- $\mathcal{N}(v)$ denotes the set of neighbors of node v,
- MEAN$\{\cdot\}$ is the neighborhood aggregation function,
- $W^{(0)}$, $W^{(1)}$ are trainable weight matrices for each layer,
- $b^{(0)}$, $b^{(1)}$ are learnable bias vectors,
- ReLU$(\cdot)$ is the non-linear activation function,
- $H_v^{(1)}$ and $H_v^{(2)}$ are the GraphSAGE embeddings after 1-hop and 2-hop aggregation, respectively.

The output embeddings from the Pass-2 layer for each node are passed through a classifier to perform the supervised node classification task. This training objective ensures that the embeddings produced by both Pass-1 and Pass-2 capture meaningful structural information for accurate classification. The classifier used in our research is a lightweight MLP layer.

3.2 Phase-2: LLM fine-tuning

In the second phase, the structure-aware node embeddings obtained from GNN are integrated into the LLM to align the eventual output embeddings based on the structural context in addition to the textual content. Instead of fine-tuning the entire LLM, we integrated LoRA only into certain layers of the model. In our setup, we introduce structural information into the LLM by integrating node representations derived from the GNN (both Pass-1 and Pass-2) directly into the middle and upper layers of the language model. The architecture is shown in Figure 2.

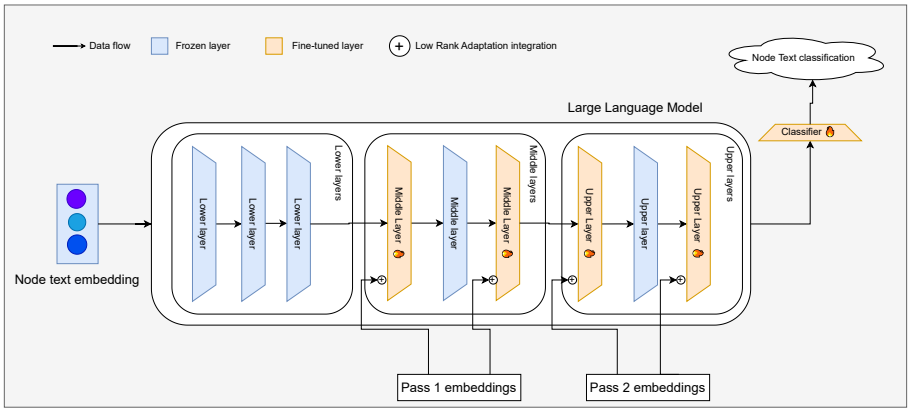

Figure 2: GaLoRA Phase-2 training

To integrate the two modalities efficiently, we use a low-rank adaptation mechanism as follows:

$$Z = W_{\mathrm{C}} \cdot (\alpha \cdot W_{\mathrm{A}} H_1 + (1 - \alpha) \cdot W_{\mathrm{B}} H_2) \tag{3}$$

where:

- $H_1 \in \mathbb{R}^{d \times T}$ represents the hidden states from the previous layer of the LLM,
- $H_2 \in \mathbb{R}^{g \times T}$ represents the structural node embeddings from the GNN broadcast across all T tokens
- $W_{\mathrm{A}} \in \mathbb{R}^{r \times d}$ projects the LLM input to a lower dimension,
- $W_{\mathrm{B}} \in \mathbb{R}^{r \times g}$ projects the graph embedding to the same low-rank space,
- $W_{\mathrm{C}} \in \mathbb{R}^{d \times r}$ projects the fused representation back to the original dimension,
- $\alpha \in [0, 1]$ is a learnable gate that balances between the text and structure inputs,
- $Z \in \mathbb{R}^{d \times T}$ is the final adapted input passed into the frozen LLM layer.
- d: dimensionality of the LLM hidden states,
- g: dimensionality of GNN output embeddings (per node),
- r: low-rank dimension used for parameter-efficient adaptation,
- T: token sequence length for each node's textual content.

This formulation allows us to inject structure-aware representations into the language model in a parameter-efficient manner, without modifying or retraining the full LLM architecture. As $r << d$ and $r << g$, where d is the dimension of input embeddings and g of graph embeddings, the number of parameters being trained is far less than that during fine-tuning the entire LLM model. The middle layers of LLM are injected with the Pass-1 embeddings and the upper layers with Pass-2 embeddings. This is based on the understanding that middle layers help form context between words, and upper layers help with context formation of higher layers.

The structural embedding from the GNN and the token embeddings from the LLM input layer are each projected into a shared lower-dimensional space of rank r, where $r \ll d$ and $r \ll g$, with d and g being the LLM layer input embedding dimension and GNN module output embedding dimension, respectively. These low-rank projections are then fused via a learnable gate and transformed to the LLM layer output dimension before being passed into the next layer. This injection mechanism enables GaLoRA to efficiently incorporate neighborhood context into the semantic understanding pipeline without retraining the full LLM, significantly reducing computational overhead.

3.3 Design choices and limitations

One of the key design choices of the framework is the injection of Pass-1 and Pass-2 embeddings in the middle and upper layers of the LLM encoder in Phase-2, respectively. This allows the LLM to be aware of the immediate neighborhood through Pass-1 embeddings while it forms an understanding of the local context of the tokens. At later stages, having a wider view of the graph through Pass-2 embeddings enables the LLM encoder to reason over a wider context during the final semantic stages.

Another important design element is the use of a learnable gate parameter. In the integration of graph embeddings in the LLM encoder, this helps the LLM regulate the structural influence on LLM for the node classifications. Despite its effectiveness, this framework is currently only evaluated for node classification tasks.

We use the cross-entropy loss in both phases of GaLoRA because it is the standard choice for node classification tasks and is used by most benchmark models. This allowed us to make fair comparisons with previous work.

We have also adopted stratified splits to balance label distributions, though this may introduce some risk of information leakage in GNNs due to message passing. Future work could mitigate this with benchmark or structure-aware splits. Further work will also explore extending GaLoRA beyond node classification to tasks such as link prediction, graph classification, and advanced fusion designs, as well as validating performance on newly proposed TAG benchmarks and diverse datasets.

4 Experimentation

4.1 Datasets

We evaluate GaLoRA on three text-attributed graph (TAG) datasets: Instagram, Reddit [9], and ArXiv [10]. Each dataset contains node-level textual content and edges representing structural relationships. Table 1 summarizes their key statistics.

Instagram: Nodes represent users with bios as text; the task is binary classification of commercial vs. non-commercial users.

Reddit: Nodes represent users with text from their last three posts; the task is binary classification of popular vs. non-popular users.

ArXiv: Nodes represent papers with titles and abstracts as text; task is 40-class paper categorization. Due to resource constraints, we evaluate on a 46K-node subgraph released by prior work.

Table 1: Summary of datasets used in experiments.

Dataset	# Nodes	# Edges	# Tokens	Split (%)	# Classes	Metric
ArXiv	46,198	78,548	35,920,710	54/18/28	40	Accuracy
Instagram	11,339	144,010	579,263	80/10/10	2	ROC-AUC
Reddit	33,434	198,448	6,748,436	80/10/10	2	Accuracy

4.2 Experimental Setup

All experiments were conducted in Google Colab using NVIDIA A100 GPUs (52 GB VRAM). Implementations utilised PyTorch [11], PyTorch Geometric [12], and HuggingFace Transformers [13] libraries, with tokenization handled by the respective pretrained model's tokenizer.

We used GraphSAGE [3] with two message passing layers to generate structure-aware node embeddings of size 64. For the language modeling component, we experimented with both GPT-2 [14] and RoBERTa [15], applying LoRA-based adaptation to 6 transformer layers (3 middle and 3 upper). The output embeddings from the GNN were aligned with the tokenized text representations and injected into the LLMs during fine-tuning.

Token lengths varied across datasets (96 for Instagram, 128 for Reddit, and 256 for ArXiv) based on the average node text length. Dataset splits and evaluation metrics are discussed in their respective sections, while training hyperparameters and ablation studies (e.g., LoRA rank) are detailed in the appendix.

4.3 Results

This section discusses the results of the method and its performance compared to baseline models. All experiments use GraphSAGE as the GNN. While prior work pairs it with Llama-13B as the pretrained language model, we use smaller LMs (GPT-2 and RoBERTa) due to computational constraints.

4.3.1 Performance evaluation

Table 2 reports a *LLM-controlled* comparison where both GaLoRA and GraphAdapter use the same PLM (RoBERTa or GPT-2). Instagram is evaluated with ROC–AUC; Reddit and ArXiv with Accuracy. We use GraphAdapter as the primary baseline, as it reports state-of-the-art results on the TAG benchmarks utilized. GaLoRA is competitive with, and often surpasses, GraphAdapter across datasets and backbones, with the largest margins on ArXiv and Instagram under GPT-2. We use a fixed stratified train/val/test split and five training seeds (0–4), reporting mean and variance across the multiple runs.

Table 2: LLM-controlled comparison (same LLM across methods). Instagram uses ROC–AUC; Reddit/ArXiv use Accuracy. For GaLoRA, results are mean ± std over 5 seeds (0–4). Baseline results (GNN, GraphAdapter) are taken directly from their reported values.

	ArXiv		**Instagram**		**Reddit**	
Model	RoBERTa	GPT-2	RoBERTa	GPT-2	RoBERTa	GPT-2
GNN (PLM)†	0.7129 (0.0013)	0.7174 (0.0019)	0.6123 (0.0063)	0.6019 (0.0124)	0.6191 (0.0043)	0.6282 (0.0036)
GraphAdapter†	**0.7273** (**0.0021**)	0.7325 (0.0022)	0.6292 (0.0033)	0.6276 (0.0034)	0.6379 (0.0061)	0.6441 (0.0022)
GaLoRA (Ours)	0.7234 (0.0014)	**0.7550** (**0.0048**)	**0.6392** (**0.0093**)	**0.6420** (**0.0046**)	**0.6464** (**0.0035**)	**0.6611** (**0.0049**)

† Baseline numbers are quoted from the GraphAdapter [6] paper; we did not retrain these models. *ArXiv:* Results use the 46k-node subgraph provided by GraphAdapter to keep experiments feasible on our hardware.

4.3.2 Parameter Efficiency Comparison

Table 3 compares pretrained language models (PLMs) and the number of trainable parameters relative to each method's own backbone size. GaLoRA trains only 0.18M parameters in the GNN (Phase 1) and 0.115M for the LoRA layers in the LLM (Phase 2), for a total of 0.295M trainable parameters (**0.238%** of GPT-2). In contrast, GLEM fine-tunes the entire DeBERTa-Large model, while GraphAdapter trains only a GNN and fusion layer (**0.015%** of LLaMA-13B) but does not fine-tune the LLM to capture semantic knowledge from the textual content. GaLoRA combines low-rank adaptation with structural embeddings, achieving competitive performance with the smallest trainable parameter footprint among methods that perform semantic fine-tuning. Note that the percentage of GaLoRA is reported with respect to GPT-2; when applied to larger models such as LLaMA-13B, the fraction of trainable parameters would be even smaller, since only a subset of layers is adapted and GNN training remains unchanged.

Table 3: Comparison of PLMs and trainable parameters. Relative % is computed w.r.t. each method's own PLM parameter count. Phase 1: GNN module; Phase 2: LoRA-injected LLM layers.

Model	PLM	# PLM Params	# Trainable Params	Relative to own PLM
GLEM	DeBERTa-Large	435M	435M	100%
GraphAdapter	LLaMA 2–13B	13B	2M	0.015%
GaLoRA (Ours)	GPT-2	124M	0.295M	**0.238%**

Summary: GaLoRA delivers competitive accuracy to state-of-the-art structural modeling approaches while training less than one percent of its LLM's parameters. Its two-phase design enables structural and semantic adaptation with minimal computational cost, making it a practical and scalable solution for resource-constrained settings.

5 Conclusion

In this work, we introduced GaLoRA (Graph-aware Low-Rank Adaptation), a modular and parameter-efficient framework for enhancing LLM performance on text-attributed graphs. By decoupling structural and semantic learning into two phases and injecting GNN-derived structural embeddings during fine-tuning, GaLoRA achieves strong classification results while significantly reducing training overhead. Our experiments demonstrate that even smaller language models like GPT-2 benefit meaningfully from structural context, highlighting GaLoRA's potential in resource-constrained settings. The framework's modular design also opens up future extensions to other graph tasks, such as link prediction or clustering, and potential exploration of richer GNN backbones or fusion strategies. Overall, GaLoRA offers a promising direction for scalable, structure-aware language model adaptation.

References

[1] Hao Yan, Chaoqi Li, Ruoyu Long, Chuxu Yan, Jie Zhao, Wenqi Zhuang, Jian Yin, Peng Zhang, Wei Han, Hong Sun, Weiqing Deng, Qianqian Zhang, Lei Sun, Xinchao Xie, and Senzhang Wang. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

[2] Bokai Jin, Guojiang Liu, Chao Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024. Early Access.

[3] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[6] Xiaorui Huang, Kai Han, Yifan Yang, Dong Bao, Qian Tao, Zhen Chai, and Qi Zhu. Can gnn be good adapter for llms? In *Proceedings of The Web Conference (WWW)*, 2024.

[7] Jiaqi Zhao, Meng Qu, Chao Li, Hao Yan, Qimai Li, Yuxiao Liu, Xing Xie, and Jian Tang. Learning on large-scale text-attributed graphs via variational inference. In *International Conference on Learning Representations (ICLR)*, 2023.

[8] Xiaoyu He, Ge Li, Shichao Zou, Jiaheng Li, Yuchi Mao, Jiajun Qian, Chengyu Gong, Dayou Li, Haozhao Zhuang, and Chang Xu. Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *International Conference on Learning Representations (ICLR)*, 2024.

[9] Xiaorui Huang, Kai Han, Yifan Yang, Dong Bao, Qian Tao, Zhen Chai, and Qi Zhu. Graphadapter datasets (instagram and reddit), 2024. Described in Section 3 of [6].

[10] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[12] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with pytorch geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[13] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP 2020: System Demonstrations*, pages 38–45, 2020.

[14] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

[15] 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. *arXiv preprint arXiv:1907.11692*, 2019.