WHEN GRAPH TOKENS SINK: A MECHANISTIC ANALYSIS OF GRAPH LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Language Models (GLMs) have become a promising direction for adapting Large Language Models (LLMs) to graph learning tasks. By transforming graph topology and node information into graph tokens, GLMs allow LLMs to jointly process structured graph inputs and textual instructions. Yet, it remains unclear how the internal behavior of these tokens relates to the task information they contain and their contribution to prediction. In this work, we analyze how LLMs process graph information through graph-token behavior in two representative GLM architectures, LLaGA and TEA-GLM, across seven LLM backbones and seven datasets.

Findings. We find that large graph-token activations do not reliably indicate greater contribution to prediction. Graph sink tokens emerge as activation-level outliers, with massive activation values along a small set of hidden-state dimensions, but do not necessarily attract the largest attention weights from query tokens. Their positional patterns differ between architectures: LLaGA sinks are overrepresented among padding tokens and rarely occur at target-node positions, whereas TEA-GLM shows no consistent positional preference across backbones and datasets. Pruning and activation patching generally affect LLaGA's node-classification predictions less when applied to sink tokens than to matched non-sink tokens. This distinction is weaker in link prediction and TEA-GLM, while swapping sink and non-sink input embeddings usually causes small accuracy changes in both architectures. Probing further shows that later LLaGA sink states contain node-label information that a simple classifier can read.

Implications. Together, these results reveal a decoupling between activation-level saliency and graph-semantic utility. Large activations, readable label information, and predictive contribution do not necessarily coincide in the same graph tokens. This distinction provides a basis for assessing graph-token construction, placement, and alignment mechanisms beyond activation magnitude alone.

1 INTRODUCTION

Graph Language Models (GLMs) have emerged as a promising framework for adapting Large Language Models (LLMs) to relational learning tasks (Wang et al., 2023; Liu et al., 2024; Chen et al., 2024b). By transforming graph topology and node features into discrete or soft "*graph tokens*," GLMs allow transformers to jointly process structured graph inputs alongside textual instructions. While these architectures show strong empirical performance across standard benchmarks (Jin et al., 2024; Ren et al., 2024), their central representational assumption remains largely unverified: that inherently non-Euclidean graph topologies can be flattened into a sequential token stream and faithfully parsed by an LLM without triggering architectural pathologies (Tang et al., 2024; Yu et al., 2026).

Recent studies from language and vision models show that Transformers often develop sink tokens, where a few tokens that attract disproportionately large attention (Xiao et al., 2023; 2024; Gu et al., 2024) or carry extremely large activations (Kovaleva et al., 2021; Wei et al., 2022; Sun et al., 2024) on a few hidden dimensions. These tokens are usually semantically weak, such as <BOS>, delimiters, or background image patches (Darcet et al., 2023; Jiang et al., 2026), and their large activations mainly shape the attention pattern instead of carrying semantic content. Similar sinks appear in vision tokens of vision language models (Kang et al., 2025; Luo et al., 2026). These findings expose a profound tension for GLMs. Graph tokens are explicitly injected to transmit vital structural

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

information, yet they are subjected to the exact same transformer dynamics known to hijack tokens for non-semantic, architectural reasons. While initial studies have noted sink-like attention patterns in graph inputs (Guan et al., 2025), a key question remains unaddressed: *when a graph token emerges with massive activations within the LLM, is the model actually utilizing it as topological reasoning for predictions, or is it merely exploiting the tokenized graph structure to build architectural artifacts?*

Present work. In this work, we study how LLMs process graph tokens through the lens of graph sink behavior by auditing the underlying activation mechanics of two representative GLM designs: LLaGA and TEA-GLM. We demonstrate that in graph modalities, 'sink' behavior manifests primarily as activation outliers; consequently, we define *graph sink tokens* by their massive magnitude along specific hidden-state dimensions. We analyze two representative architectures, LLaGA and TEA-GLM, across seven LLM backbones and seven datasets on node classification and link prediction. We first characterize graph sink tokens through their activation dimensions, positions, and query-to-graph attention. We then use pruning, swapping, and activation patching to compare their effects on prediction with those of matched non-sink tokens. Finally, we perform label probing and logit-lens analysis to examine the information available in their hidden states. Together, these analyses show that activation magnitude, readable task information, and predictive contribution are distinct properties of graph-token representations.

2 PRELIMINARIES

Definition of Graph Sink Tokens. Prior works showed that some tokens in Transformer-based models emerge as attention, feature-norm, or activation outliers in language and vision tasks. Specifically, Xiao et al. (2023; 2024) demonstrated that Transformers tend to assign a significant amount of attention weights to initial tokens, which are known as "attention sinks". In vision encoders like Vision Transformers, Darcet et al. (2023) also showed that some tokens, named as "artifacts", emerge as high-norm tokens in less semantic meaningful patches within the image. Sun et al. (2024) further showed the correlation of massive activations with attention sinks, where specific hidden dimensions ("sink dimensions") of the sink tokens produce unusually large activation values during the residual stream. Similar activation-based definitions of sink tokens have been adopted and extended to vision-language models by Kang et al. (2025); Luo et al. (2026) in identifying vision sink tokens.

Following Sun et al. (2024); Kang et al. (2025); Luo et al. (2026), let a GLM take a mixed-token sequence containing both text tokens and graph tokens. We denote the set of graph-token indices by $\mathcal{I}_g$, and let $\mathbf{x}_j^l \in \mathbb{R}^d$ be the hidden state of graph token j at layer l, where d is the LLM hidden dimension. The set of graph sink tokens at layer l is defined as

$$\widehat{\mathcal{I}}_g^l = \left\{ x_j \in \mathcal{I}_g \;\middle|\; \phi\left(\mathbf{x}_j^{l-1}\right) \geq \tau \right\}, \tag{1}$$

where $\phi(\cdot)$ is a sink characteristic function and τ is a predefined activation threshold: Given a set of sink dimensions $\mathcal{D}_{\text{sink}}$, we define

$$\phi\left(\mathbf{x}_j^{l-1}\right) = \max_{\bar{d} \in \mathcal{D}_{\text{sink}}} \left| \mathrm{RMSNorm}\left(\mathbf{x}_j^{l-1}\right)_{\bar{d}} \right|. \tag{2}$$

Here, $\bar{d}$ denotes one such sink dimension, and $\mathrm{RMSNorm}$ denotes Root Mean Square Normalization. Prior work has shown that sink dimensions can be language-model specific (Sun et al., 2024; Kang et al., 2025; Luo et al., 2026; Sun et al., 2026). For example, LLaMA2-7B (Touvron et al., 2023) has been shown to contain sink dimensions such as $\{1415, 2533\}$. In this paper, we study the sink effects on different LLM backbone families with different sizes, including Vicuna, LLaMA (Touvron et al., 2023; Grattafiori et al., 2024), Qwen (Bai et al., 2023), and Yi (Young et al., 2024).

GLM Choices. We study two representative GLMs: LLaGA (Chen et al., 2024a) and TEA-GLM (Wang et al., 2024). They represent two common ways of converting graph information to an LLM. LLaGA uses a node-aligned graph-token sequence, where we use its Neighborhood Detail (ND) template: the center node is placed at graph sequence index 0 and its k-hop neighborhood is organized into a fixed-shape tree by sampling neighbors at each hop. In their original paper the authors choose a 2-hop neighborhood ($k = 2$). If the sampled neighborhood is smaller than the budget, `[PAD]` tokens are inserted to maintain graph-token length. TEA-GLM follows a different design. It first uses a GNN encoder to aggregate neighborhood structure into a central-node representation and then projects this representation into a fixed number of learnable graph-token embeddings through a linear projector. These graph tokens are aligned with the LLM hidden space through contrastive learning

against principal components of LLM token embeddings. We choose these models for two reasons: i) they represent two broad GLM families: node-aligned graph tokens and encoder-embedded graph tokens; and ii) both models use fixed-length graph-token sequences, which allows us to compare graph sink positions across samples, where K is the number of graph tokens.

Experiment Setup. We study graph sink effects on both node classification (NC) and link prediction (LP) tasks across seven LLM backbones mentioned above and across seven datasets: three citation networks (Arxiv (Hu et al., 2020), Cora (Yang et al., 2016), PubMed (Sen et al., 2008)), and four E-commerce datasets from the TAG benchmark (Yan et al., 2023) – Computer (Ele-Computer), History (Book-History), Photo (Ele-photo), and Sports (Sports-Fitness). For each GLM architecture, we first use 300 calibration graphs to identify the sink dimensions D_{sink} for each dataset-backbone pair, using Eq. 2. We perform this calibration once per pair and keep the resulting dimensions fixed during evaluation. Following Eq. 1 and Eq. 2, we set the sink threshold $\tau_{\text{dim}} = 15$ for selecting qualified sink dimensions for a given graph token x_j. After the sink dimensions are selected, we set $\tau_{\text{tok}} = 20$ in selecting qualified graph sink tokens for each layer. We fix the settings for all backbones, datasets, graph tasks, and the two GLMs.

3 Graph Sink Tokens Emerge as Activation-Level Outliers

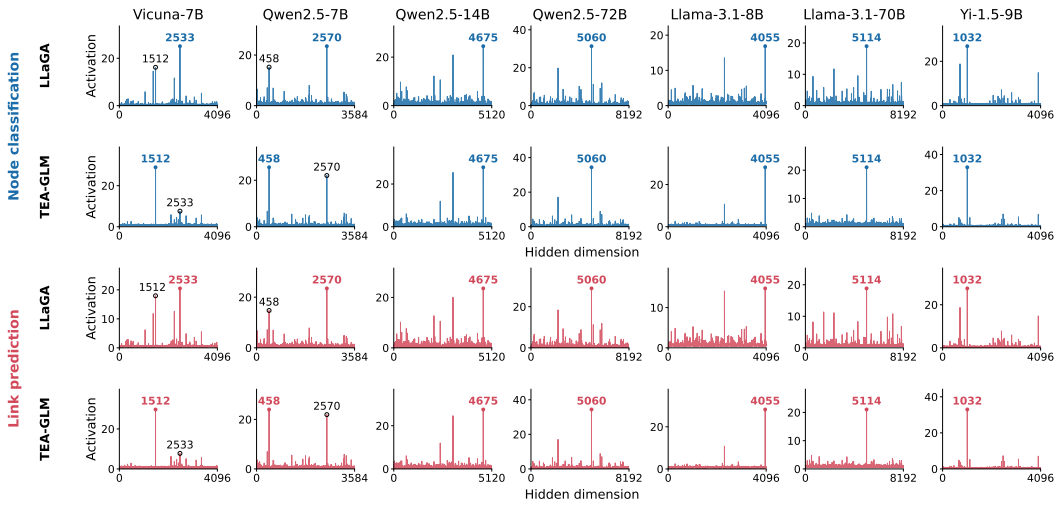


Figure 1: **Graph-sink activation magnitudes on Arxiv.** Mean RMS-Norm absolute activations across hidden dimensions for detected sink tokens in LLaGA and TEA-GLM, using seven LLM backbones. The upper panels show NC and the lower panels show LP task. A colored marker shows each panel's largest mean activation magnitude. When the two GLMs have different largest sink dimension in magnitude, we mark the activation of the other GLM's largest sink dimension for the same backbone and task in black open marker. We take the mean of layers and 300 LLaGA and TEA-GLM evaluation examples.

Graph Sink Activation Magnitudes. We examine and take the average of the activation of hidden dimensions from layer 2 to layer -2. We choose layers 2 and -2 since Sun et al. (2026) showed that the activation pattern of Transformer-based LLMs exhibits "step-up" and "step-down" blocks, where its magnitude starts to spike near the beginning and consistently disappears towards the end of the layers. Figure 1 shows the average activation magnitudes of graph sink tokens on Arxiv. They are computed from 300 evaluation samples per dataset and task for LLaGA and TEA-GLM. Results for other datasets are provided in Appendix A.1. Across backbones and tasks, most hidden dimensions have low activation magnitudes, while the identified sink dimensions exhibit massive activation spikes. For each architecture-backbone pair, the dominant sink dimensions and activation magnitudes remain largely consistent across datasets as well as node classification and link prediction tasks. The sink dimensions can differ between GLM architectures: Vicuna-7B peaks at dimension 2533 in LLaGA and 1512 in TEA-GLM for Arxiv, while Qwen2.5-7B peaks at dimensions 2570 and 458, respectively. Thus, the concentration of large activations in sink dimensions is consistent across datasets and

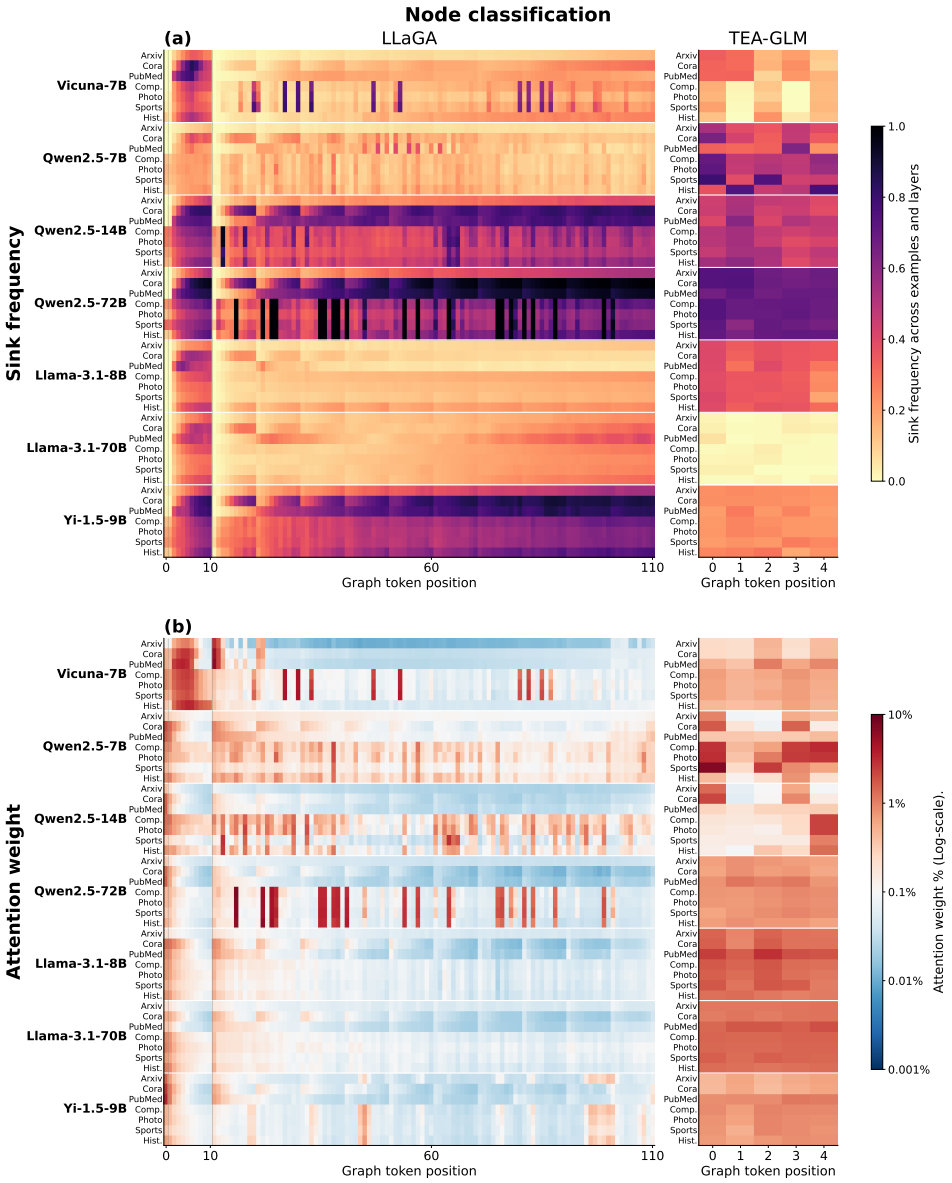


Figure 2: **Graph sink frequency and attention weight by graph token position on Node Classification tasks.** Left: LLaGA with 111 token positions; right: TEA-GLM with 5 token positions. Each row represents a pair of LLM backbone and a dataset. Subplot (a) shows the frequency of sink distribution across examples. Subplot (b) denotes the query-to-graph attention weight as percentages represented on a logarithmic color scale.

downstream graph tasks, although the specific dimensions depend on the backbone and different encoding methods of graph modality by GLM architectures can have slight impact on their emergence.

Positional Distributions of Graph Sink Tokens. To locate these outliers, we examine sink frequency at each graph token position for both GLMs. LLaGA uses the ND template, with one center-node at index 0, ten first-hop slots and 100 second-hop slots. The center node is the classification target for NC or an LP endpoint. From index 11 onward, each consecutive group of ten slots contains neighbors sampled from the corresponding first-hop node, and any unfilled node slots are filled by [PAD]. NC uses one such subgraph template ($K = 111$) while LP uses two ($K = 222$). TEA-GLM uses five graph tokens for both tasks, where each token is projected from the GNN-encoder generated embeddings.

Fig. 2-(a) shows the positional heatmaps for graph sink distributions for both GLMs. As shown, TEA-GLM sinks can occur across all five indices, with no consistent preferred position across backbones

Table 1: **Padding (%) among LLaGA sinks for citation networks.** Each entry here shows the percentage of detected sink tokens located at the [PAD] positions for NC and LP tasks.

Backbone	Arxiv	Cora	PubMed
Vicuna-7B	100.0/100.0	100.0/100.0	100.0/100.0
Qwen2.5-7B	97.4/97.4	98.9/98.4	99.9/99.7
Qwen2.5-14B	81.5/70.5	97.8/97.0	96.8/92.3
Qwen2.5-72B	100.0/99.9	100.0/100.0	100.0/100.0
Llama-3.1-8B	100.0/100.0	100.0/100.0	100.0/100.0
Llama-3.1-70B	100.0/100.0	100.0/100.0	100.0/100.0
Yi-1.5-9B	88.4/84.3	99.9/99.9	99.8/99.6
Input PAD	33.4/27.0	78.2/75.4	75.3/63.5

Table 2: **LLaGA sinks (%) on hop-views for citation networks.** Sink percentage at center/first-hop/second-hop positions, averaged over NC and LP tasks. The graph tokens in first and second hops include both [PAD] and actual graph tokens that represent node information.

Backbone	Arxiv	Cora	PubMed
Vicuna-7B	0.0/19.7/80.3	0.0/18.8/81.2	0.0/20.4/79.6
Qwen2.5-7B	0.1/11.4/88.5	0.0/12.7/87.3	0.0/11.5/88.5
Qwen2.5-14B	0.0/9.1/90.9	0.0/8.0/92.0	0.0/8.3/91.7
Qwen2.5-72B	0.0/8.0/92.0	0.0/7.3/92.7	0.0/7.7/92.3
Llama-3.1-8B	0.0/17.0/83.0	0.0/27.1/72.9	0.0/33.1/66.9
Llama-3.1-70B	0.0/12.1/87.9	0.0/14.4/85.6	0.0/13.1/86.9
Yi-1.5-9B	0.1/7.4/92.5	0.0/6.9/93.1	0.0/7.4/92.6

and datasets. LLaGA exhibits a more structured distribution including repeated bands and blocks. For Vicuna, Qwen2.5-7B, and the two LLaMA models, sinks emerge sparsely across the entire graph sequence. Whereas the distribution of the remaining three models significantly extend beyond the beginning of the sequence. We also find that sink tokens are less likely to emerge at the center node or first-hop nodes starting from token position 11, 21, onward, which is why the figures contains clear separated blocks for every 10 indices. However, note that due to graph size differences, not every center node contains exactly 10 first-hop nodes at position 1 and can be filled with [PAD] instead.

We further try to quantify whether graph sink tokens are semantically less meaningful using LLaGA as an example. Table 1 shows that [PAD] tokens account for 70.5-100.0% of detected sinks on the citation datasets. For Vicuna and both Llama-3.1 backbones, the percentages of sinks being [PAD] are 100%. This shows that activation outliers frequently occur on tokens without node-attribute information. This [PAD] over-representation also holds on the E-commerce datasets, although several backbones produce larger portion of sinks on real graph tokens (App. A.3). Table 2 reports the mean percentage of sink position distributions on the center, first and second hop nodes, evaluated on LLaGA on the citation datasets. No sink tokens emerge on the center node in 19 of the 21 entries and to 0.1% in the remaining two. First-hop tokens contribute 6.9%-33.1% of sinks, significantly less than the second-hop tokens.

Attention Weight relationship with Graph Sink Tokens. We next compare the sink frequencies with query-to-graph attention weights in Figure 2-(b). In LLaGA, the center node at position 0 receives relatively high attention weights in most cases, despite its low sink rate. It also shows that first-hop nodes that are non-[PAD] tokens receive relatively higher attention weights at position 1. This shows that LLMs can assign high attention to meaningful graph tokens such as the target node token. Some sink tokens do appear to receive high attention weights in E-commerce dataset for Vicuna and Qwen2.5-14B and 72B models, but this is not a consistent feature of graph sinks. In TEA-GLM, the attention weight received by sink tokens differs across backbones. Together, these results show that sink tokens identified by massive activations do not necessarily become attention sinks. This is consistent with the findings with Sun et al. (2026), where they show that massive activations and attention sinks can be suppressed separately.

4 Role of Graph Sink Tokens in Prediction

In Sec. 3, we show that graph sink tokens are activation-level outliers, yet they do not necessarily receive disproportionately high attention weights. In this section, we examine their role in prediction through token pruning and token swapping interventions, followed by mechanistic analyses for each GLM architecture. For LLaGA, we study how label information develops in sink states and whether replacing those states changes predictions. For TEA-GLM, we use logit lens to probe the vocabulary predictions from the five graph tokens.

4.1 Intervention Experiments

Intervention setup. For sink PRUNING, we remove graph token hidden states from the sequence before each transformer block. At each block, we apply Eq. 1 with $\tau_{\text{tok}} = 20$ to the identified graph

sink tokens in the current intervened sequence and select up to two qualified sinks with the largest activation magnitudes. We remove their entire hidden-state vectors and pass the shortened sequence to the block. The remaining tokens retain their original position IDs. We repeat this procedure at every block, so sink selection responds to the intervened states and deletions accumulate over depth. If no surviving graph token satisfies the sink criterion, no token is removed at that block. As control experiments, we prune randomly selected non-sink graph tokens, matching the number of sink-token deletions at each block. For sink SWAPPING, we select the two sink tokens that are most consistently selected across all blocks during a non-intervened run and exchange its input embeddings with two randomly selected non-sink graph tokens. We use five different seeds for random non-sink deletion and swapping interventions. We report the NC and LP results on citation networks in Table 3, and the remaining E-commerce results are in App A.4. We note that following the original work of TEA-GLM, we train TEA-GLM's projectors separately on Arxiv and Computer, and use the projectors to perform zero-shot predictions on the remaining datasets. LP task for TEA-GLM is also evaluated zero-shot from the pretrained Arxiv and Computer projectors, respectively. This setup will cause several configurations to degenerate to constant or near-constant predictions. We explicitly mark those cases in the table.

Main Results. The results in Table 3 show that pruning sink tokens has little effect on prediction accuracy in most settings, whereas pruning random non-sink tokens often causes a large performance drop in LLaGA NC settings. For instance, Qwen2.5-72B on PubMed retains its 98.00% accuracy after sink pruning but drops to 34.27% when pruning random non-sink tokens as control. The same results also holds for the E-commerce datasets in Table 8, the median changes from baseline across all 49 configurations are 0.0 and -15.7% for pruning sink tokens and random non-sink tokens, respectively. The differences between pruning sink and controlled non-sink tokens are less drastic in LP. On the citation networks, sink and non-sink pruning change accuracy by medians of -0.3 and -1.9%. TEA-GLM is less sensitive to both pruning: across the total 49 NC configurations, the median accuracy changes are 0.0 and 0.1 percentage points for sink and random non-sink pruning. We note that repeated pruning in this case can eventually remove all five TEA-GLM graph tokens. Pruning sinks can even improve prediction accuracy: Qwen2.5-14B on Cora LP increases from 59.00% to 65.67%. We exclude the marked constant or near-constant LP predictors when interpreting token importance, since unchanged accuracy in those cases does not demonstrate useful prediction. TEA-GLM also receives node text in its prompt, which provides task-relevant information alongside the graph tokens. We therefore hypothesize that both sink and non-sink graph tokens provide limited additional benefit over these text inputs in the tested settings, and that graph token representations can sometimes interfere with prediction under zero-shot transfer. Graph information may already have reached the text-token states before all graph tokens are removed.

Swapping sink tokens with random non-sink tokens also has little effect on prediction for both GLMs. Across all NC configurations for each GLM, LLaGA and TEA-GLM change by at most 1.0%, with a median change of 0.0 for both models. In LP tasks, the median change is -0.3% for both models after excluding the flagged configurations in TEA-GLM. The largest absolute changes are 1.8% for LLaGA and 2.4% for TEA-GLM. Swapping interventions preserve the input embeddings but move them to different graph token positions. These results suggest that the original positions of the selected sink tokens are usually not essential for prediction.

4.2 MECHANISTIC ANALYSIS

The interventions results above measure how removing or moving graph tokens changes prediction. However, these results leave open whether sink tokens contain little task-relevant information or whether the information they contain contributes little to prediction. We examine this distinction through activation patching in both GLMs, and we perform label probing in LLaGA and logit-lens analysis in TEA-GLM.

Activation Patching. We use activation patching to test whether graph token states carry information that the model uses for prediction (Heimersheim & Nanda, 2024). We pair two examples for which the unmodified model predicts different classes. At a selected transformer block, we replace the hidden states of selected graph tokens in the first example with the states at the same positions in the second example, keeping all other states unchanged. We then continue the forward pass and measure how much the output shifts toward the class predicted for the second example. Each intervention patches only one block, and we compare sink-token patches with patches of the same number of non-sink tokens. We also patch all graph token positions to test the effect of replacing the full graph representation.

Table 3: **Graph token interventions on citation networks.** We report the intervention results of sink and random nonsink token pruning and sink swapping experiments. Every entry shows the prediction accuracy and we compare the interventions against the baseline performance. At each transformer block, we prune at most two sink tokens with the highest activation magnitudes. Non-sink token pruning matches the number of block-wise pruned sink tokens. Swap exchanges the selected input embeddings. Results marked in red show non-sink pruning yields lower accuracy in sink pruning. For TEA-GLM LPs, † shows NC-to-LP zero-shot transfer and ‡ shows constant or near-constant predictions and their interventions are marked in gray and thus neglected.

		Node classification (NC)			Link prediction (LP)		
Backbone	Intervention	Arxiv	Cora	PubMed	Arxiv	Cora	PubMed
LLaGA							
Vicuna-7B	Baseline	77.33	87.00	95.33	92.00	84.40	87.20
	Prune sinks	77.33	86.33	95.33	91.33	84.80	87.60
	Non-sink	68.13±0.88	66.47±2.33	71.53±1.59	90.87±0.45	82.29±0.29	87.60±0.91
	Swap	76.93±0.53	87.20±0.27	94.80±0.34	91.40±0.25	84.88±0.90	88.64±0.59
Qwen2.5-7B	Baseline	60.67	82.00	91.00	66.33	79.33	83.67
	Prune sinks	62.00	78.33	93.33	66.67	79.67	85.00
	Non-sink	60.87±0.27	77.20±0.62	91.40±1.06	66.67±0.37	77.80±0.75	84.67±0.60
	Swap	61.80±0.27	79.47±0.27	91.47±0.16	67.13±0.54	78.40±0.49	82.80±0.50
Qwen2.5-14B	Baseline	72.67	87.00	97.67	78.00	83.67	91.00
	Prune sinks	72.33	87.00	97.67	78.33	78.33	80.00
	Non-sink	60.40±0.74	46.07±1.44	50.00±0.76	78.00±0.76	71.00±0.94	85.00±0.47
	Swap	73.00±0.00	86.07±0.13	95.07±0.44	78.33±0.00	84.40±0.13	89.20±0.34
Qwen2.5-72B	Baseline	70.00	86.00	98.00	78.33	83.00	88.67
	Prune sinks	71.00	85.67	98.00	77.67	82.00	88.33
	Non-sink	48.80±0.45	35.80±0.34	34.27±0.77	70.27±0.71	60.67±0.76	72.13±0.62
	Swap	67.53±0.54	85.73±0.39	95.07±0.83	78.13±0.27	82.80±0.72	87.73±0.57
Llama-3.1-8B	Baseline	67.33	85.33	92.33	78.67	82.67	89.00
	Prune sinks	67.00	85.33	92.33	79.00	82.33	90.33
	Non-sink	67.20±0.93	84.00±0.87	91.00±0.67	79.80±0.16	80.93±0.39	88.93±0.49
	Swap	67.33±0.21	85.60±0.25	92.20±0.34	77.93±0.53	82.40±0.33	88.73±0.39
Llama-3.1-70B	Baseline	70.33	84.67	89.00	80.33	83.67	89.67
	Prune sinks	70.33	85.33	89.00	79.33	83.67	86.00
	Non-sink	59.87±0.72	74.13±0.91	67.60±1.14	79.47±0.40	81.53±0.81	86.93±1.20
	Swap	70.07±0.39	84.73±0.13	89.13±0.16	80.27±0.39	82.93±0.57	87.87±0.54
Yi-1.5-9B	Baseline	74.33	87.67	98.33	80.67	85.00	89.00
	Prune sinks	72.67	87.33	97.33	78.33	84.00	87.33
	Non-sink	55.60±1.24	34.67±1.45	42.80±1.67	78.73±1.02	59.20±1.51	67.40±1.27
	Swap	72.73±0.39	87.27±0.25	98.20±0.16	80.20±0.16	83.67±0.70	88.80±0.16
TEA-GLM							
Vicuna-7B	Baseline	58.00	13.33	78.67	57.67†	62.00†	58.33†
	Prune sinks	57.67	13.33	80.33	57.67	62.00	57.67
	Non-sink	57.80±0.50	13.60±0.13	80.13±0.16	57.33±0.30	61.53±0.50	57.60±0.25
	Swap	57.67±0.00	13.67±0.00	78.67±0.00	56.20±0.16	60.07±0.25	57.67±0.21
Qwen2.5-7B	Baseline	59.33	24.33	85.00	49.67†‡	50.67†‡	46.00†‡
	Prune sinks	59.67	24.67	86.00	49.67	50.67	46.00
	Non-sink	59.67±0.37	24.80±0.16	85.20±0.16	49.67±0.00	50.67±0.00	46.00±0.00
	Swap	59.00±0.00	24.33±0.00	85.33±0.00	49.67±0.00	50.67±0.00	46.00±0.00
Qwen2.5-14B	Baseline	66.33	33.67	85.67	55.00†	59.00†	77.00†
	Prune sinks	65.67	33.00	85.67	59.00	65.67	80.00
	Non-sink	65.93±0.88	31.00±0.76	85.40±0.13	57.87±0.86	63.87±0.78	79.27±0.93
	Swap	66.67±0.00	33.00±0.00	85.33±0.00	55.33±0.00	58.33±0.00	77.33±0.00
Qwen2.5-72B	Baseline	60.33	30.00	84.33	50.33†‡	51.00†‡	46.33†‡
	Prune sinks	61.33	30.33	84.67	50.67	50.67	46.33
	Non-sink	60.73±0.71	29.53±0.34	84.40±0.25	50.33±0.00	51.40±0.25	46.33±0.00
	Swap	60.33±0.00	30.67±0.00	84.33±0.00	50.33±0.00	51.33±0.00	46.33±0.00
Llama-3.1-8B	Baseline	39.67	15.00	70.00	49.67†‡	50.67†‡	46.00†‡
	Prune sinks	46.00	12.00	68.67	49.67	50.67	46.00
	Non-sink	39.40±0.39	14.00±0.37	69.60±0.13	49.67±0.00	50.67±0.00	46.00±0.00
	Swap	39.33±0.00	15.00±0.00	69.67±0.00	49.67±0.00	50.67±0.00	46.00±0.00
Llama-3.1-70B	Baseline	44.00	9.67	71.67	49.00†	50.00†	57.33†
	Prune sinks	43.67	10.33	72.00	50.00	52.67	57.33
	Non-sink	43.13±0.27	10.00±0.00	71.33±0.37	49.20±0.50	51.27±0.25	57.53±0.50
	Swap	44.07±0.77	10.47±0.54	70.07±0.53	49.87±0.27	49.33±0.21	54.93±1.39
Yi-1.5-9B	Baseline	50.33	4.00	69.33	49.67†‡	51.33†‡	46.00†‡
	Prune sinks	50.67	4.33	69.33	49.67	51.33	46.00
	Non-sink	50.53±0.45	4.00±0.21	70.27±0.13	49.67±0.00	51.33±0.00	46.00±0.00
	Swap	53.33±0.00	4.00±0.00	69.67±0.00	49.67±0.00	51.33±0.00	46.00±0.00

Let c_r and c_d denote the classes predicted for the first and second examples, and let p_r, p_d, and p_{patch} be the output class distributions for the two unmodified examples and the patched first example. The log-probability difference $m(p) = \log p(c_d) - \log p(c_r)$ measures how much the model predicts the second example's class over the first example's class. We measure the change caused by patching as

$$T = \frac{m(p_{\text{patch}}) - m(p_r)}{m(p_d) - m(p_r)}. \tag{3}$$

Here, $T = 0$ means that patching leaves this class preference unchanged, while $T = 1$ means that it reaches the preference observed for the second example. Table 4 shows median T values across eligible configurations with seven backbones.

Table 4: **Effects of Activation Patching.** Median transfer ratios T across eligible backbone configurations. Sink and non-sink patch an equal number of tokens. Whole graph block patches all graph token positions. Control setting copies each example's own hidden states as control experiment. Constant or near-constant configurations without sinks are excluded.

	LLaGA		TEA-GLM	
Patched positions	NC	LP	NC	LP
Control	0.000	0.000	0.000	0.000
Sink	0.008	0.004	0.001	0.000
Non-sink	0.041	0.005	0.000	0.002
Whole graph	0.651	0.539	0.005	0.019

Table 5: **Logistic Regression accuracy (%) at predicting node labels from LLaGA token states.** We report the median logistic regression probe accuracy across all backbones on Arxiv NC. Columns show token embeddings used to fit the regression model at projector output (Input), middle/final block inputs (Middle/Last), and final normalized output (Final) levels.

	Representation Depth			
Token set	Input	Middle	Last	Final
Sinks	20.7	61.0	62.7	62.5
Random nodes	58.9	63.4	63.9	63.1
Target (Center) node	69.7	69.6	67.4	67.4
All graph tokens	65.4	62.7	64.1	63.7

Table 4 shows that replacing LLaGA sink states has less effect on NC predictions than replacing the same number of non-sink states. Non-sink patches give larger values in 47 of 49 configurations. Replacing all graph token states produces a much larger median T. Thus, LLaGA uses information from its graph token states for NC, but the selected sink states have a smaller effect than the matched non-sink states. The difference is less clear in LP, where sink and non-sink patches both have small median T values. Conversely, the effect of activation patching is less significant in TEA-GLM for both sink and non-sink patches. Even replacing all five graph token states gives median T values of only 0.005 in NC and 0.019 in LP. These results agree with the small effects of pruning in Section 4. The small differences between TEA-GLM's sink and non-sink patching scores therefore do not provide a reliable ranking of token importance.

Predicting node labels from LLaGA token states. Patching measures how much token states affect the model's output. We next test whether a simple classifier can predict the target node's label from graph token representations at different stages of the LLM's forward pass. LLaGA's template lets us compare sink tokens with tokens representing the target node and its sampled neighbors. We extract different token hidden states using $3{,}000$ Arxiv NC examples across seven backbones at the projector output, the inputs to the middle and final transformer blocks, and the output after final normalization. For each backbone, token set, and depth, we fit a separate logistic-regression classifier using the mean of the RMS-normalized token states as features and the target-node labels as targets. The LLM remains unchanged. We report accuracy on held-out folds using five-fold cross validation.

Table 5 shows that predicting node labels from sink states becomes easier at later layers. At the input, median classifier accuracy is 20.7% for sinks, compared with 58.9% for random graph nodes tokens (non-PAD tokens) and 69.7% for the target node. At the final state, sink accuracy reaches 62.5%, close to the random-node control at 63.1%, while the target node remains higher at 67.4%. Thus, a simple classifier can predict node labels from later sink states, even though the sink interventions usually have little effect on the LLM's NC predictions.

Logit-lens analysis of TEA-GLM. TEA-GLM compresses the graph into five learned tokens. We apply the language model's output head to each slot's hidden state at different layers and inspect the vocabulary token with the highest probability. This logit-lens analysis provides a vocabulary readout of each state. We choose the top2 sink positions based on the frequency of the indices having the most number of sinks across all transformer blocks. Figure 3 shows that Vicuna-7B on PubMed

dataset produces dataset-level words such as *paper* and *the*, token fragments, and punctuation. It also shows repeated semantic meaningless tokens across all five tokens in the middle and later layers. These readouts do not clearly identify the target classes. The same pattern appears in the selected sink slots and the other graph slots, including cases where the generic vocabulary token has a relatively high probability. We show additional Arxiv and Cora examples in Appendix A.5.

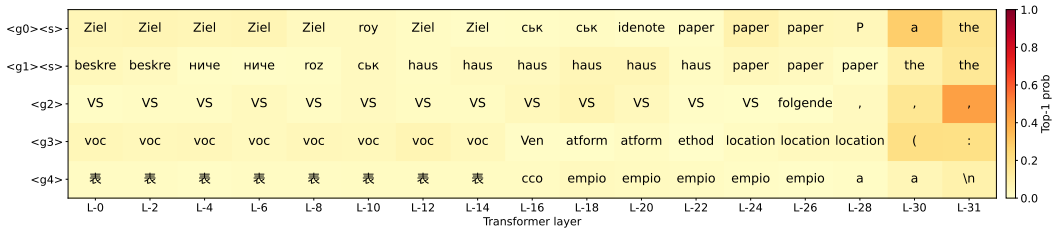


Figure 3: **Logit-lens readouts of TEA-GLM on PubMed.** Top-one vocabulary predictions across the five learned graph slots and transformer layers. Color indicates prediction probability. We select the sink positions based on the frequency across layers. Generic words, token fragments, punctuation, and newline-containing tokens appear across sink and other positions.

5 Related Work

Graph Language Models. Recent work has increasingly explored how LLMs can be adapted to graph-structured data, especially for text-attributed graphs, citation networks, recommendation graphs, and molecular graphs (Li et al., 2023; Jin et al., 2024; Ren et al., 2024; Zhang et al., 2024; 2025a; Yu et al., 2026). Existing approaches can be roughly grouped by the role assigned to the LLM: some use LLMs as predictors over verbalized graph prompts (Guo et al., 2023; Ye et al., 2024; Chen et al., 2024a; Ye et al., 2024), some use LLMs as text encoders to enrich node features, and others align graph representations with the LLM embedding space for generative or instruction-following graph tasks (Wang et al., 2024; Zhao et al., 2023; Tang et al., 2024; Liu et al., 2024). These methods show that LLMs can provide useful semantic priors for graph learning (Wang et al., 2026).

Attention Sinks. Attention sinks have been widely observed in Transformer-based models. In LLMs, attention sinks often occur at the first position within the sequence and are often semantic meaningless (Xiao et al., 2024; Gu et al., 2024; Barbero et al., 2025; An et al., 2025; Liu et al., 2026). These attention sinks receive disproportionate attention from other tokens. Related work on massive activations shows that sink behavior is often accompanied by extreme hidden-state values on a small number of feature dimensions (Dettmers et al., 2022; Sun et al., 2024; Oh et al., 2024). Recent studies further show that attention sinks are not limited to LLMs, but also appear in vision-language and multimodal models. In vision encoders like vision transformers (ViTs), there are high feature norm tokens located in background image patches (Darcet et al., 2023; Lu et al., 2025; Jiang et al., 2026; Fesser et al., 2026). In vision language models, vision tokens projected from ViT also exhibit sink-like behavior with model and task-dependent effect (Kang et al., 2025; Zhang et al., 2025b; Cappellazzo et al., 2025; Luo et al., 2026).

6 Conclusion

We study activation defined graph sink tokens in LLaGA and TEA-GLM across seven LLM backbones and seven datasets. We find that their large activations do not consistently co-occur with high query-to-graph attention. Graph sinks exhibit large activations in a few hidden dimensions, but do not consistently receive the highest query-to-graph attention. Controlled interventions show that LLaGA node classification generally depends more on non-sink states than sink states, while this distinction is weaker in link prediction and TEA-GLM. Label probing further shows that later LLaGA sink states contain task information despite their smaller effects on prediction. Together, these findings separate three properties of graph-token representations: activation magnitude, readable label information, and predictive contribution. Our study shows that activation magnitude alone is insufficient to assess graph-token importance and demonstrates how controlled interventions and information probes can explain their different roles.

AI DISCLOSURE

In this work, we used generative AI tools to assist with code implementation and debugging, improve the plotting and presentation of the main figures and main tables, as well as supporting manuscript writing, editing, and proofreading. Generative AI was not used to frame the main research idea or design the core research methodology. We have reviewed all AI-assisted work: all AI-generated code was verified and tested by the authors, and all AI-suggested text and figure edits were checked against the experimental results and center statements before inclusion. The authors take full responsibility for the final content of this paper, including all text, claims, and codes produced with the aid of generative AI.

ETHICS STATEMENT

This work studies how GLMs process graph information using existing citation and E-commerce benchmarks and pretrained language models. It does not involve human-subject experiments or the collection of new personal data. The datasets and pretrained models may contain biases that affect their representations and predictions. Our experiments examine graph-token behavior and predictive contribution, but they do not evaluate the fairness or safety of these models in deployment. We report differences across architectures, backbones, and tasks, and identify configurations with constant or near-constant predictions to make the scope of the findings clear.

REPRODUCIBILITY STATEMENT

The main text defines the sink-detection criteria, describes the graph-token interventions and matched controls, and specifies the activation-patching and probing analyses. Appendix D documents the LLM backbones, graph-token construction, node features, training procedures, and hyperparameters. We use a fixed training seed and report variation across five seeds for the main randomized intervention controls. Appendix A provides additional results, while Appendix B examines the token threshold and includes rank-based analyses without a token cutoff. We also distinguish supervised training from zero-shot evaluation, particularly for TEA-GLM, and mark configurations with constant or near-constant predictions when reporting intervention results. We will release the code upon publication.

REFERENCES

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Systematic outliers in large language models. In *International Conference on Learning Representations*, volume 2025, pp. 31483–31504, 2025.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. Why do llms attend to the first token? *arXiv preprint arXiv:2504.02732*, 2025.

Umberto Cappellazzo, Stavros Petridis, Maja Pantic, et al. Mitigating attention sinks and massive activations in audio-visual speech recognition with llms. *arXiv preprint arXiv:2510.22603*, 2025.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*, 2024a.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61, 2024b.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35: 30318–30332, 2022.

Keyu Duan, Qian Liu, Tat-Seng Chua, Shuicheng Yan, Wei Tsang Ooi, Qizhe Xie, and Junxian He. Simteg: A frustratingly simple approach improves textual graph learning. *arXiv preprint arXiv:2308.02565*, 2023.

Lukas Fesser, Mozes Jacobs, Thomas Fel, Andy Keller, and Sham Kakade. A unifying view of attention sinks: Two algorithms, two solutions. *arXiv preprint arXiv:2606.08105*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*, 2024.

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.

Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*, 2023.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Nicholas Jiang, Amil Dravid, Alexei Efros, and Yossi Gandelsman. Vision transformers don't need trained registers. *Advances in neural information processing systems*, 38:56557–56595, 2026.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 36 (12):8622–8642, 2024.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. *arXiv preprint arXiv:2503.03321*, 2025.

Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. Bert busters: Outlier dimensions that disrupt transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3392–3405, 2021.

Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*, 2023.

Yiming Liu, Bin Lu, Xinbing Wang, Chenghu Zhou, and Meng Jin. Slash the sink: Sharpening structural attention inside llms. *arXiv preprint arXiv:2605.10503*, 2026.

Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V Chawla. Can we soft prompt llms for graph learning tasks? In *Companion proceedings of the ACM web conference 2024*, pp. 481–484, 2024.

Andrew Lu, Wentinn Liao, Liuhui Wang, Huzheng Yang, and Jianbo Shi. Artifacts and attention sinks: Structured approximations for efficient vision transformers. *arXiv preprint arXiv:2507.16018*, 2025.

Jiayun Luo, Wan-Cyuan Chris Fan, Lyuyang Wang, Xiangteng He, Tanzila Rahman, Purang Abolmaesumi, and Leonid Sigal. To sink or not to sink: Visual information pathways in large vision-language models. In *International Conference on Learning Representations*, volume 2026, pp. 93626–93660, 2026.

Jaehoon Oh, Seungjun Shin, and Dokwan Oh. House of cards: Massive weights in llms. *arXiv preprint arXiv:2410.01866*, 2024.

Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh Chawla, and Chao Huang. A survey of large language models for graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6616–6626, 2024.

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, 2008.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

Shangwen Sun, Alfredo Canziani, Yann LeCun, and Jiachen Zhu. The spike, the sparse and the sink: Anatomy of massive activations and attention sinks. *arXiv preprint arXiv:2603.05498*, 2026.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 491–500, 2024.

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.

Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. *Advances in neural information processing systems*, 37:5950–5973, 2024.

Haoyu Wang, Peihao Wang, Mufei Li, Shikun Liu, Siqi Miao, Zhangyang Wang, Pan Li, et al. Graph-kv: Breaking sequence via injecting structural biases into large language models. *Advances in Neural Information Processing Systems*, 38:68358–68387, 2026.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36:30840–30861, 2023.

Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. Outlier suppression: Pushing the limit of low-bit transformer language models. *Advances in Neural Information Processing Systems*, 35:17402–17414, 2022.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pp. 38087–38099. PMLR, 2023.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. *arXiv preprint arXiv:2410.10819*, 2024.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, et al. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264, 2023.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Language is all a graph needs. In *Findings of the association for computational linguistics: EACL 2024*, pp. 1955–1973, 2024.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

Shuo Yu, Yingbo Wang, Ruolin Li, Guchun Liu, Yanming Shen, Shaoxiong Ji, Bowen Li, Fengling Han, Xiuzhen Zhang, and Feng Xia. Graph2text or graph2token: A perspective of large language models for graph learning. *ACM Transactions on Information Systems*, 44(3):1–49, 2026.

Delvin Ce Zhang, Menglin Yang, Rex Ying, and Hady W Lauw. Text-attributed graph representation learning: Methods, applications, and challenges. In *Companion Proceedings of the ACM Web Conference 2024*, pp. 1298–1301, 2024.

Ding Zhang, Jane Downer, Can Chen, and Ren Wang. Efficient and robust continual graph learning for graph classification in biology. *IEEE Transactions on Signal and Information Processing over Networks*, 2025a.

Stephen Zhang, Mustafa Khan, and Vardan Papyan. Attention sinks: A'catch, tag, release'mechanism for embeddings. *arXiv preprint arXiv:2502.00919*, 2025b.

Haiteng Zhao, Shengchao Liu, Ma Chang, Hannan Xu, Jie Fu, Zhihong Deng, Lingpeng Kong, and Qi Liu. Gimlet: A unified graph-text model for instruction-based molecule zero-shot learning. *Advances in neural information processing systems*, 36:5850–5887, 2023.

A Additional Analysis

A.1 Massive Activation Magnitudes on More Datasets

In this subsection, we provide additional analysis and figures on massive activation magnitudes on the remaining two citation network datasets and four E-commerce datasets in Fig 4 - Fig 9. Similar to the phenomena shown in Fig 1, sink tokens exhibit massive activation magnitudes in sparse sink dimensions. Those sparse sink dimensions are fixed and consistent across backbones regardless of inputs, datasets, or tasks.

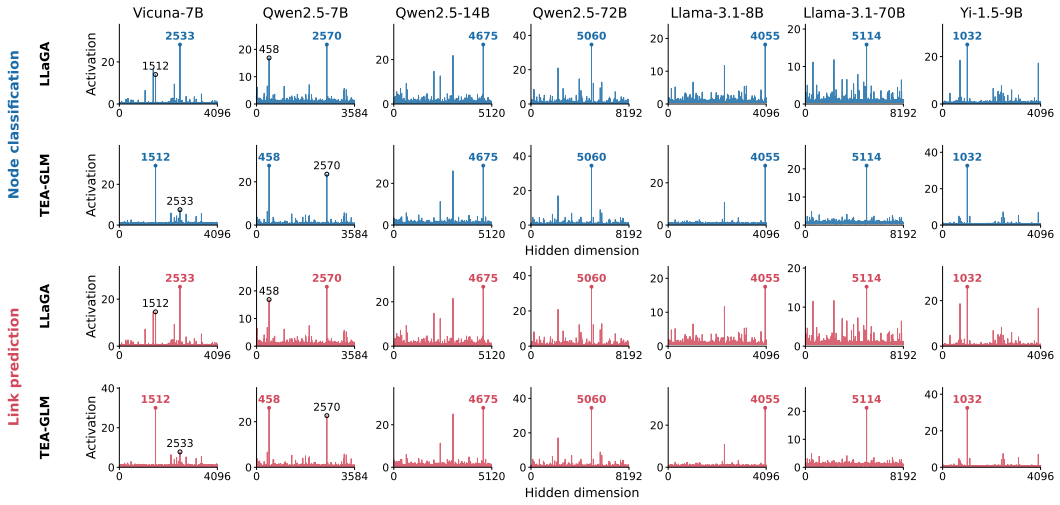


Figure 4: **Graph-sink activation magnitudes on Cora.** Mean RMS-Norm absolute activations across hidden dimensions for detected sink tokens in LLaGA and TEA-GLM, using seven LLM backbones. The upper panels show NC and the lower panels show LP task. A colored marker shows each panel's largest mean activation magnitude. When the two GLMs have different largest sink dimension in magnitude, we mark the activation of the other GLM's largest sink dimension for the same backbone and task in black open marker. We take the mean of layers and 300 LLaGA and TEA-GLM evaluation examples.

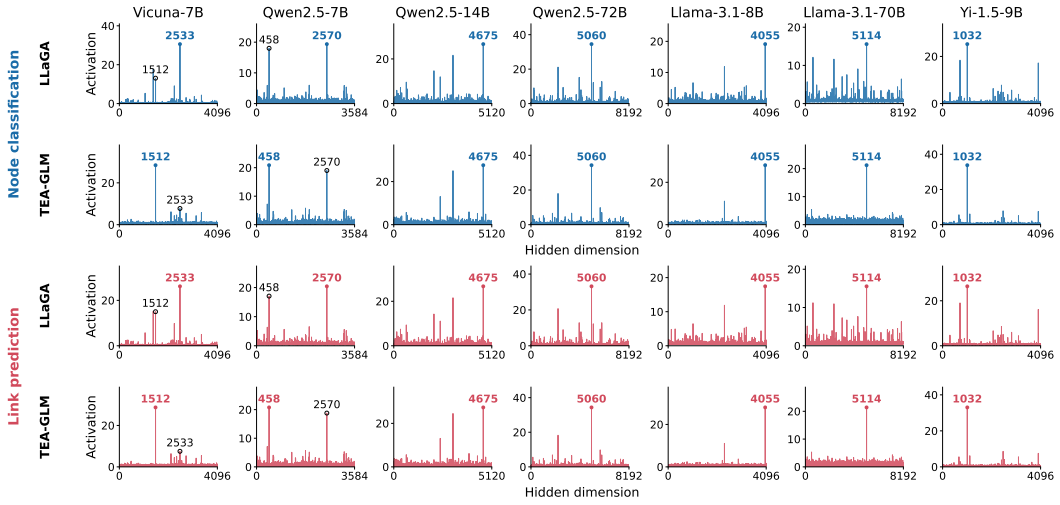


Figure 5: **Graph-sink activation magnitudes on Pubmed.** Mean RMS-Norm absolute activations across hidden dimensions for detected sink tokens in LLaGA and TEA-GLM, using seven LLM backbones. The upper panels show NC and the lower panels show LP task. A colored marker shows each panel's largest mean activation magnitude. When the two GLMs have different largest sink dimension in magnitude, we mark the activation of the other GLM's largest sink dimension for the same backbone and task in black open marker. We take the mean of layers and 300 LLaGA and TEA-GLM evaluation examples.

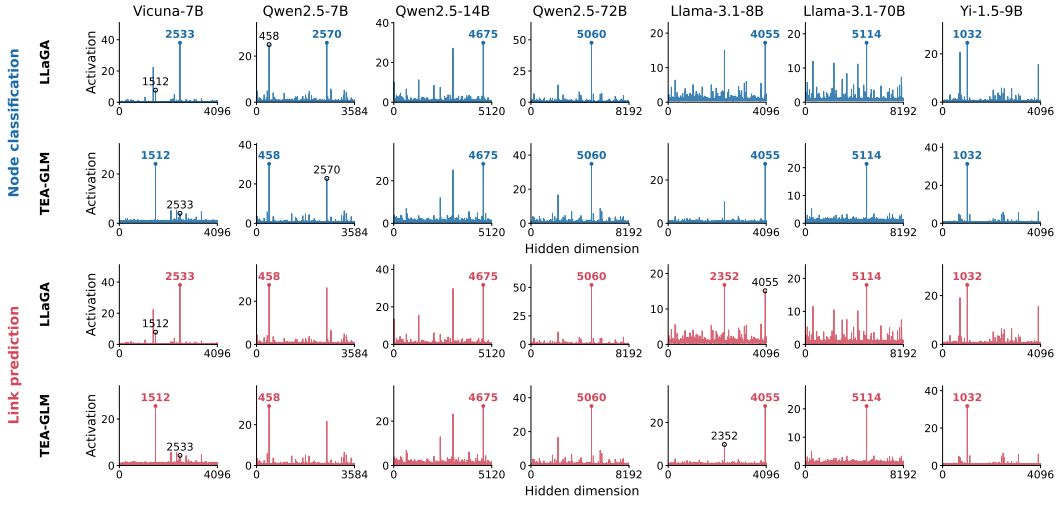


Figure 6: **Graph-sink activation magnitudes on Computer.** Mean RMS-Norm absolute activations across hidden dimensions for detected sink tokens in LLaGA and TEA-GLM, using seven LLM backbones. The upper panels show NC and the lower panels show LP task. A colored marker shows each panel's largest mean activation magnitude. When the two GLMs have different largest sink dimension in magnitude, we mark the activation of the other GLM's largest sink dimension for the same backbone and task in black open marker. We take the mean of layers and 300 LLaGA and TEA-GLM evaluation examples.

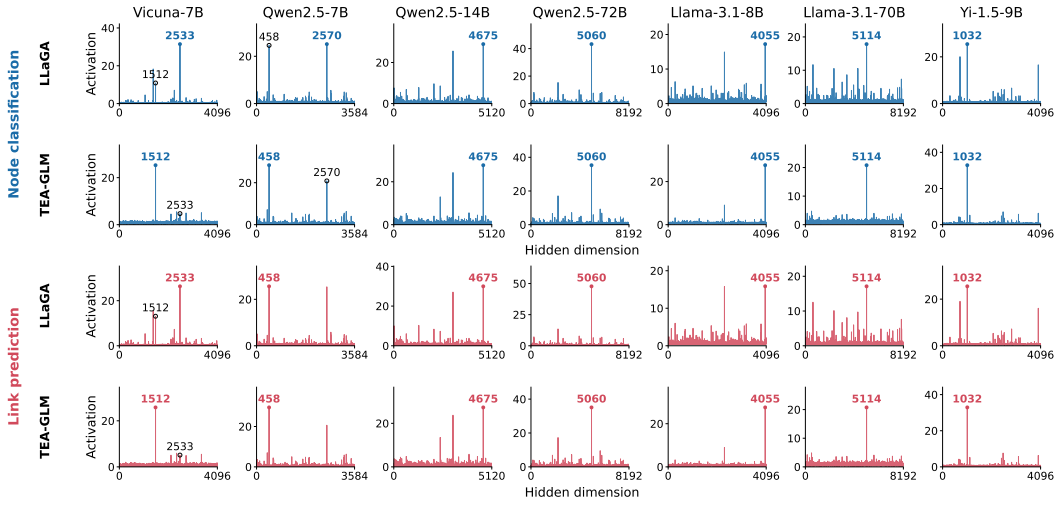


Figure 7: **Graph-sink activation magnitudes on History.** Mean RMS-Norm absolute activations across hidden dimensions for detected sink tokens in LLaGA and TEA-GLM, using seven LLM backbones. The upper panels show NC and the lower panels show LP task. A colored marker shows each panel's largest mean activation magnitude. When the two GLMs have different largest sink dimension in magnitude, we mark the activation of the other GLM's largest sink dimension for the same backbone and task in black open marker. We take the mean of layers and 300 LLaGA and TEA-GLM evaluation examples.

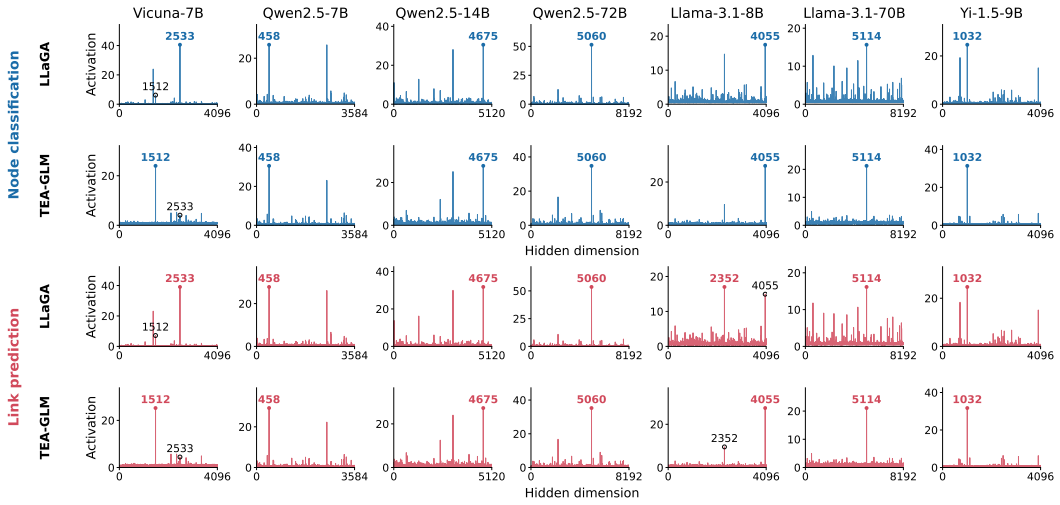


Figure 8: **Graph-sink activation magnitudes on Photo.** Mean RMS-Norm absolute activations across hidden dimensions for detected sink tokens in LLaGA and TEA-GLM, using seven LLM backbones. The upper panels show NC and the lower panels show LP task. A colored marker shows each panel's largest mean activation magnitude. When the two GLMs have different largest sink dimension in magnitude, we mark the activation of the other GLM's largest sink dimension for the same backbone and task in black open marker. We take the mean of layers and 300 LLaGA and TEA-GLM evaluation examples.

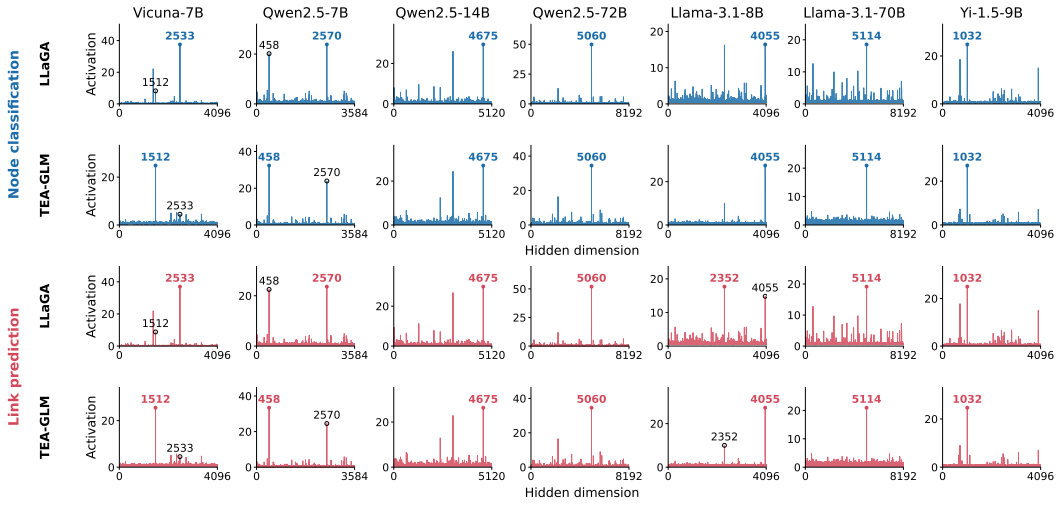


Figure 9: **Graph-sink activation magnitudes on Sports.** Mean RMS-Norm absolute activations across hidden dimensions for detected sink tokens in LLaGA and TEA-GLM, using seven LLM backbones. The upper panels show NC and the lower panels show LP task. A colored marker shows each panel's largest mean activation magnitude. When the two GLMs have different largest sink dimension in magnitude, we mark the activation of the other GLM's largest sink dimension for the same backbone and task in black open marker. We take the mean of layers and 300 LLaGA and TEA-GLM evaluation examples.

A.2 Graph Sink Positions and Attention in Link Prediction

Figure 10 extends the positional analysis in Figure 2 to LP. The results show broadly similar patterns to NC. In LLaGA, the two endpoint subgraphs exhibit similar sink distributions, with repeated bands following the ND template. The figures show that sink frequency varies across backbones and datasets: Qwen2.5-14B, Qwen2.5-72B, and Yi-1.5-9B show frequent sinks across many positions on Cora and PubMed, while several other backbones show sparser patterns. These frequent sink positions do not consistently receive high attention. For example, Qwen2.5-72B has high sink frequencies across much of both endpoint sequences on Cora and PubMed, but relatively low attention at many of those positions. In TEA-GLM, sinks occur across the five slots without a consistent preferred position across settings. Llama-3.1-70B shows particularly low sink frequencies while still assigning attention to its graph slots. Together, results for LP task generally agree with the NC results: large graph-token activations and high query-to-graph attention do not consistently occur at the same positions in the sequence.

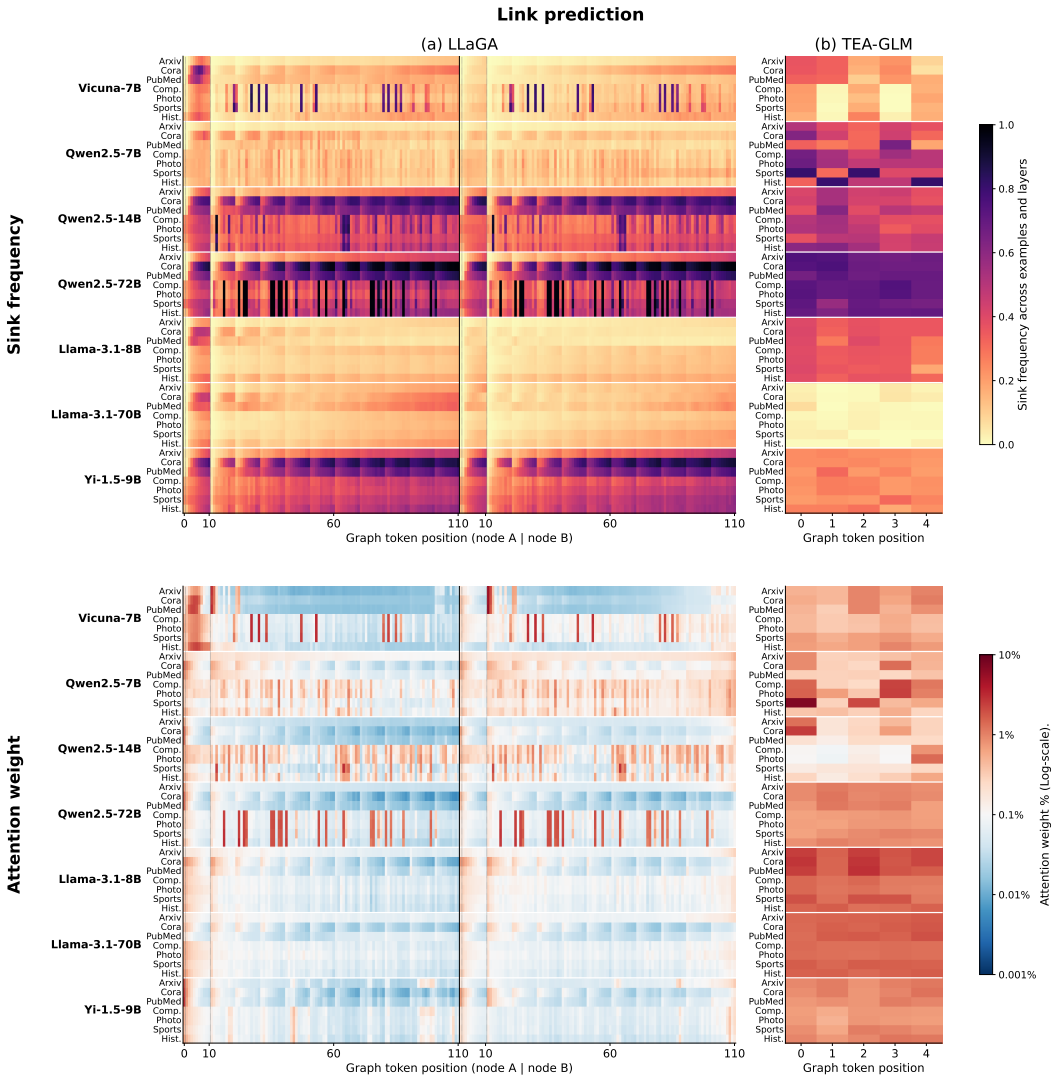


Figure 10: **Graph sink frequency and attention weight by graph token position on Link Prediction tasks.** Left: LLaGA with 222 token positions; right: TEA-GLM with 5 token positions. Each row represents a pair of LLM backbone and a dataset. Subplot (a) shows the frequency of sink distribution across examples. Subplot (b) denotes the query-to-graph attention weight as percentages represented on a logarithmic color scale.

A.3 Graph Sink Roles and Positions on E-Commerce Datasets

We report the percentage of LLaGA sinks that are PAD tokens and their distribution across template positions on the e-commerce datasets in Tables 6 and 7. Across all four datasets and seven backbones, the PAD percentage of sink tokens exceeds its percentage among input graph tokens in both NC and LP. Several backbones have more sinks on real graph tokens than in the citation datasets. For example, PAD tokens account for 45.2% and 30.9% of Qwen2.5-7B sinks on Photo for NC and LP, respectively, compared with 97.4-99.9% on the citation datasets. For both Llama-3.1 backbones, nearly all sinks remain PAD tokens. The positional distributions are also similar to those in Table 2. Averaged equally over NC and LP, center positions account for only 0.0-0.9% of sinks, while first-hop and second-hop positions account for 7.1-15.2% and 84.2-92.7%, respectively.

Table 6: Padding (%) among LLaGA sinks for e-commerce networks. Each entry here shows the percentage of detected sink tokens located at the [PAD] positions for NC and LP tasks.

Backbone	Computers	Photo	Sports	History
Vicuna-7B	73.2 / 48.8	58.0 / 44.7	60.7 / 48.3	93.1 / 85.6
Qwen2.5-7B	56.5 / 38.3	45.2 / 30.9	48.5 / 42.1	67.8 / 62.8
Qwen2.5-14B	60.1 / 38.5	55.9 / 36.2	63.3 / 50.9	76.6 / 66.5
Qwen2.5-72B	69.0 / 51.2	68.2 / 51.8	59.7 / 49.4	80.3 / 68.5
Llama-3.1-8B	100.0 / 100.0	99.9 / 100.0	100.0 / 100.0	100.0 / 100.0
Llama-3.1-70B	100.0 / 100.0	100.0 / 100.0	100.0 / 100.0	100.0 / 100.0
Yi-1.5-9B	84.0 / 69.7	76.0 / 63.6	71.7 / 63.3	89.5 / 82.9
Input PAD	45.1 / 27.0	38.9 / 25.6	37.3 / 28.5	52.5 / 39.3

Table 7: LLaGA sinks (%) on hop-views for e-commerce datasets. Sink percentage at center/first-hop/second-hop positions, averaged over NC and LP tasks. The graph tokens in first and second hops include both [PAD] and actual graph tokens that represent node information.

Backbone	Computers	Photo	Sports	History
Vicuna-7B	0.1 / 11.3 / 88.6	0.1 / 10.9 / 89.0	0.2 / 9.4 / 90.4	0.6 / 15.2 / 84.2
Qwen2.5-7B	0.8 / 12.1 / 87.1	0.9 / 12.0 / 87.1	0.9 / 11.0 / 88.1	0.5 / 11.3 / 88.2
Qwen2.5-14B	0.5 / 9.3 / 90.2	0.6 / 9.1 / 90.3	0.5 / 8.9 / 90.6	0.3 / 8.6 / 91.1
Qwen2.5-72B	0.4 / 7.6 / 92.0	0.3 / 7.1 / 92.6	0.5 / 7.7 / 91.8	0.2 / 7.1 / 92.7
Llama-3.1-8B	0.0 / 14.0 / 86.0	0.0 / 14.0 / 86.0	0.0 / 13.9 / 86.1	0.0 / 13.2 / 86.8
Llama-3.1-70B	0.0 / 12.9 / 87.1	0.0 / 11.2 / 88.8	0.0 / 10.9 / 89.1	0.0 / 11.8 / 88.2
Yi-1.5-9B	0.3 / 8.0 / 91.7	0.4 / 8.3 / 91.3	0.5 / 8.2 / 91.3	0.2 / 7.6 / 92.2

A.4 GRAPH-TOKEN INTERVENTIONS ON E-COMMERCE DATASETS

We report the NC and LP results of sink pruning, matched non-sink pruning, and sink swapping on the four e-commerce datasets in Table 8, extending Table 3. For LLaGA NC, pruning non-sink tokens generally reduces accuracy more than pruning sinks. Across the 28 backbone-dataset settings, the median accuracy changes are -0.2% and -15.5% for sink and non-sink pruning, respectively. For example, Yi-1.5-9B retains its 74.00% accuracy on History after sink pruning, while non-sink pruning reduces it to 40.20%. The distinction is less consistent in LP, where either intervention can increase or decrease accuracy. TEA-GLM shows smaller changes under both pruning interventions in NC, and neither consistently causes larger drops in the unflagged LP settings. Swapping also produces relatively small changes in both architectures. As in Table 3, we exclude the marked constant or near-constant LP predictors when interpreting these effects. These results extend the main finding to the e-commerce datasets: the difference between sink and non-sink pruning is most evident in LLaGA NC.

Table 8: **Graph-token interventions on the E-commerce networks.** Entries are accuracy (%); subscripts give standard deviations over five seeds. Prune sinks and Non-sink use adaptive physical pruning; Swap exchanges selected input embeddings. Red marks lower non-sink than sink-pruning accuracy in unflagged configurations. Symbols on LP baseline cells identify †: an NC-trained projector used zero-shot; ‡: constant or near-constant predictions in the patching diagnostics (one class accounts for at least 95% of examples). All four LP entries for a flagged configuration are gray and are omitted from LP summaries.

		Node classification (NC)				Link prediction (LP)			
Backbone	Intervention	Computers	Photo	Sports	History	Computers	Photo	Sports	History
LLaGA									
Vicuna-7B	Baseline	92.00	89.33	92.00	73.67	55.67†	47.00†	50.67†	50.67†‡
	Prune sinks	92.67	87.67	91.67	73.33	54.00	50.00	50.67	50.67
	Non-sink	90.00±0.42	87.07±0.68	90.80±0.62	73.53±0.81	55.33±0.56	45.73±0.57	50.80±0.62	50.67±0.00
	Swap	92.07±0.25	89.40±0.25	92.00±0.00	73.27±0.33	55.20±0.34	46.20±0.45	50.33±0.21	50.67±0.00
Qwen2.5-7B	Baseline	90.00	82.33	91.33	69.33	59.67†	61.00†	51.33†	63.67†
	Prune sinks	90.67	81.00	91.33	67.00	62.33	61.67	51.67	69.33
	Non-sink	90.60±0.49	80.67±0.60	91.40±0.57	67.53±0.96	59.27±0.61	59.67±1.05	51.20±0.34	62.60±0.57
	Swap	90.20±0.34	81.27±0.33	91.47±0.16	67.67±0.52	60.27±0.68	59.87±0.54	50.80±1.13	62.73±1.22
Qwen2.5-14B	Baseline	91.00	83.67	91.33	71.33	56.67†‡	48.33†‡	51.00†‡	50.67†‡
	Prune sinks	88.33	77.33	92.00	70.33	57.00	49.00	51.00	50.67
	Non-sink	66.47±0.54	67.07±1.14	80.47±0.62	49.87±0.34	57.27±1.00	51.93±0.68	53.67±0.21	55.40±0.49
	Swap	90.67±0.00	85.00±0.00	91.67±0.00	71.00±0.00	56.33±0.00	48.40±0.13	51.00±0.00	50.67±0.00
Qwen2.5-72B	Baseline	93.33	89.33	93.33	74.00	57.00†‡	49.33†‡	51.00†‡	50.67†‡
	Prune sinks	87.67	85.33	89.67	69.67	57.00	49.33	51.00	50.67
	Non-sink	72.07±0.39	61.00±0.21	69.67±0.70	51.13±1.00	58.73±0.33	52.33±0.56	61.73±0.39	58.60±0.39
	Swap	93.33±0.00	89.33±0.00	93.33±0.00	74.33±0.00	57.00±0.00	49.33±0.00	51.00±0.00	50.67±0.00
Llama-3.1-8B	Baseline	93.00	87.67	93.00	73.00	55.33†‡	47.67†‡	50.00†‡	50.67†‡
	Prune sinks	93.00	87.33	93.00	73.00	55.67	48.00	50.33	50.67
	Non-sink	88.73±0.83	81.67±1.10	92.13±0.58	70.33±0.47	55.07±0.13	47.40±0.39	50.07±0.13	50.67±0.00
	Swap	93.00±0.42	87.67±0.00	93.00±0.00	73.00±0.37	55.20±0.16	47.40±0.13	49.53±0.27	50.67±0.00
Llama-3.1-70B	Baseline	93.67	86.00	94.00	71.33	60.00†	50.00†	57.67†	47.67†
	Prune sinks	93.67	86.67	94.00	71.33	58.33	52.00	57.00	46.33
	Non-sink	78.40±0.98	65.87±0.54	74.33±0.60	55.60±1.10	59.73±0.88	52.67±1.01	62.47±0.96	53.33±0.56
	Swap	93.60±0.13	85.80±0.27	93.80±0.16	71.93±0.25	60.27±0.68	51.07±0.93	57.20±0.34	46.67±0.21
Yi-1.5-9B	Baseline	92.33	87.00	92.00	74.00	51.33†	44.00†	42.67†	51.67†‡
	Prune sinks	92.00	87.67	92.00	74.00	51.33	40.67	41.00	51.33
	Non-sink	59.80±0.98	54.60±0.74	61.40±0.74	40.20±0.72	46.93±1.14	43.73±1.73	39.47±1.17	57.80±0.58
	Swap	92.33±0.00	87.33±0.00	92.00±0.00	74.00±0.00	51.33±0.00	44.00±0.00	42.67±0.00	51.67±0.00
TEA-GLM									
Vicuna-7B	Baseline	39.33	35.67	25.67	37.67	57.67†	56.33†‡	61.67†	63.67†
	Prune sinks	39.33	36.00	25.67	39.00	56.67	56.33	64.00	63.00
	Non-sink	39.93±0.25	36.00±0.00	26.47±0.27	38.40±0.25	56.07±0.57	56.47±0.16	62.93±0.33	62.80±0.69
	Swap	39.93±0.33	35.47±0.34	26.33±0.56	38.73±0.13	56.27±0.25	57.33±0.30	62.20±0.58	62.73±0.39
Qwen2.5-7B	Baseline	46.67	46.67	42.67	27.33	50.00†	54.00†	56.67†‡	48.00†‡
	Prune sinks	47.00	44.33	42.33	27.00	49.00	52.67	55.00	48.00
	Non-sink	46.40±0.13	45.80±0.27	43.53±0.16	27.00±0.00	50.33±0.00	53.73±0.13	56.27±0.13	48.00±0.00
	Swap	47.00±0.00	46.67±0.00	43.00±0.00	27.33±0.00	49.67±0.00	53.67±0.00	57.33±0.00	48.00±0.00
Qwen2.5-14B	Baseline	47.33	54.00	51.00	21.33	52.67†	57.33†	81.00†	86.67†
	Prune sinks	48.33	53.00	50.00	21.67	52.33	56.33	80.33	87.33
	Non-sink	47.73±0.39	53.53±0.78	50.07±0.25	21.27±0.13	52.27±0.33	57.47±0.40	81.07±0.33	86.07±0.39
	Swap	46.33±0.00	54.00±0.00	50.33±0.00	20.67±0.00	52.33±0.00	57.00±0.00	81.33±0.00	86.33±0.00
Qwen2.5-72B	Baseline	48.33	57.00	43.00	31.33	49.67†‡	53.67†	54.33†‡	48.00†‡
	Prune sinks	46.33	55.33	41.67	34.00	51.00	53.33	54.00	48.00
	Non-sink	46.93±0.61	57.47±0.34	42.80±0.34	31.87±0.45	49.93±0.39	53.40±0.33	55.00±0.00	48.00±0.00
	Swap	48.33±0.00	57.33±0.00	42.67±0.00	31.67±0.00	50.00±0.00	53.33±0.00	54.33±0.00	48.00±0.00
Llama-3.1-8B	Baseline	43.33	21.67	33.00	7.33	47.00†‡	49.00†‡	52.67†‡	48.00†‡
	Prune sinks	40.67	23.00	31.67	7.00	47.00	49.00	52.67	48.00
	Non-sink	44.73±0.44	22.13±0.34	32.60±0.49	6.53±0.34	47.00±0.00	49.00±0.00	52.67±0.00	48.00±0.00
	Swap	44.67±0.00	21.33±0.00	33.33±0.00	7.33±0.00	47.00±0.00	49.00±0.00	52.67±0.00	48.00±0.00
Llama-3.1-70B	Baseline	24.00	10.67	26.67	19.33	52.33†	53.00†	52.00†	54.00†
	Prune sinks	26.33	10.00	27.00	19.33	51.67	51.33	52.00	55.00
	Non-sink	25.33±0.37	9.80±0.16	27.00±0.30	18.80±0.27	51.80±0.62	51.00±0.76	52.27±0.49	55.53±0.88
	Swap	25.00±0.76	10.33±0.21	26.93±0.25	19.07±0.65	51.53±0.54	53.80±1.07	50.13±0.50	52.73±0.93
Yi-1.5-9B	Baseline	42.33	47.00	39.33	14.33	46.00†‡	53.00†	53.67†‡	50.33†‡
	Prune sinks	43.67	48.00	39.33	15.00	46.00	52.67	53.33	49.33
	Non-sink	45.00±0.42	48.20±0.40	39.40±0.44	14.93±0.39	45.47±0.16	53.27±0.13	53.07±0.25	50.20±0.16
	Swap	43.00±0.00	48.33±0.00	39.33±0.00	14.67±0.00	46.00±0.00	52.67±0.00	53.33±0.00	50.33±0.00

A.5 Additional Logit Lens Results

We report additional logit-lens results for TEA-GLM on Arxiv and Cora datasets in Figures 11 and 12. The logit-lens readouts show a similar pattern to the examples in the main text: many graph-token states decode to subword fragments, generic words, or punctuation. On both datasets, the sink slots repeatedly decode to `paper` in later layers. Other slots produce broad terms such as `type` and, on Cora, `category`. At the final displayed layer, commas and newline tokens appear across several slots, including non-sink slots.

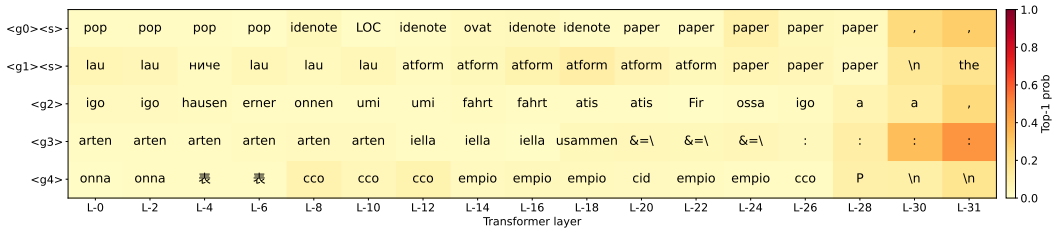


Figure 11: **Logit-lens readouts of TEA-GLM graph on Arxiv.** Top-one vocabulary predictions across the five learned graph slots and transformer layers. Color indicates prediction probability. We select the sink positions based on the frequency across layers. Generic words, token fragments, punctuation, and newline-containing tokens appear across sink and other positions.

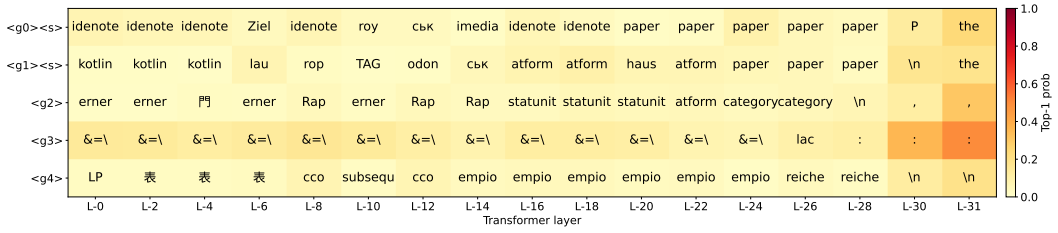


Figure 12: **Logit-lens readouts of TEA-GLM on Cora.** Top-one vocabulary predictions across the five learned graph slots and transformer layers. Color indicates prediction probability. We select the sink positions based on the frequency across layers. Generic words, token fragments, punctuation, and newline-containing tokens appear across sink and other positions.

B SENSITIVITY ANALYSIS

In this section, we justify the value choice τ, specifically τ_{dim} and τ_{tok} of sink characteristic function to select qualified graph sink tokens. We examine how the two thresholds changes sink detection and intervention results, and support this analysis with token rankings that do not require τ_{tok}. We report the seven backbones and seven datasets used in the main experiments. We use 300 samples for both settings, evaluated at the inputs to interior blocks $2, \ldots, L-2$. Unless stated otherwise, we average datasets and backbones to acquire the final results.

B.1 CHOICE OF DETECTION THRESHOLDS

The dimension threshold τ_{dim} selects the coordinates used to score graph tokens, whereas τ_{tok} determines which tokens qualify as sinks. In the main text, we use $\tau_{\text{dim}} = 15$ to select sink dimensions with large activations. Decreasing this threshold relaxes the selection criterion and allows more dimensions to qualify as sink dimensions. A token's score is the largest absolute value of its RMS-normalized hidden state over the selected dimensions.

Table 9 reports the results for $\tau_{\text{tok}} \in 5, 10, 20, 30, 50$. With $\tau_{\text{tok}} = 5$, most graph tokens meet the sink criterion: on average, 91.8 of 111 tokens in LLaGA NC, 177.2 of 222 in LLaGA LP, and approximately 3.8 of 5 in TEA-GLM. At $\tau_{\text{tok}} = 20$, these counts decrease to 36.0, 59.8, and approximately 1.7, respectively, while 94.5-99.2% of examples still contain a sink in at least one evaluated block. Increasing the threshold to 50 leaves only 40.4-44.2% of LLaGA examples and approximately 17% of TEA-GLM examples with a detected sink. We therefore use $\tau_{\text{tok}} = 20$ to avoid counting most graph tokens as sinks while retaining sinks in most examples.

Table 9: Sink detection under different token thresholds. Each entry shows the mean number of sinks and transformer block (from block 2 to -2), followed by the percentage of examples with at least one sink in any interior block. Values are averaged over the seven main backbones and seven datasets. Bold indicates the threshold used in the main experiments.

τ_{tok}	5	10	**20**	30	50
LLaGA NC	91.8 (100.0)	64.6 (97.5)	**36.0 (94.5)**	18.7 (93.3)	2.7 (44.2)
LLaGA LP	177.2 (100.0)	114.1 (98.8)	**59.8 (96.4)**	31.5 (95.5)	5.0 (40.4)
TEA-GLM NC	3.8 (100.0)	2.9 (100.0)	**1.7 (98.4)**	1.0 (81.4)	0.04 (17.2)
TEA-GLM LP	3.8 (100.0)	2.9 (100.0)	**1.7 (99.2)**	1.0 (83.3)	0.03 (17.3)

The main positional patterns also persist around the selected threshold. Relative to $\tau_{\text{tok}} = 20$, the mean overlap between sink-position histograms is 0.86-0.91 at thresholds 10 and 30 across the two architectures and tasks, where 1 denotes identical distributions; settings with no detected sinks are omitted from this overlap statistic. For LLaGA, the mean share of sink occurrences on `[PAD]` slots remains 80.8-86.1% in NC and 73.1-80.3% in LP over thresholds 10, 20, and 30. These results support the positional findings without requiring identical token sets at each threshold.

B.2 THRESHOLD-FREE SENSITIVITY ANALYSIS

To check whether the observations depend on a particular τ_{tok}, we rank graph tokens by their activation scores within each example and block, using the same sink dimensions without applying τ_{tok}. Table 10 compares the highest token score with the median and reports where high-scoring tokens occur. For LLaGA, the mean highest-to-median score ratio ranges from 1.75 to 4.35 in NC and from 2.00 to 4.85 in LP across backbones. Padding tokens generally score above real-node tokens, with `[PAD]`-versus-real-node AUROC values of 0.85-1.00. The target node has a mean rank percentile of at most 0.36, where 1 denotes the highest score. These results show that LLaGA's high activation scores are associated with padding rather than target-node positions even without a token threshold. TEA-GLM has smaller highest-to-median ratios, indicating smaller differences between the highest and median scores among its five graph tokens.

We then exclude the two highest-scoring graph tokens at every block without requiring their scores to exceed $\tau_{\text{tok}} = 20$. Table 11 reports accuracy changes using 300 examples per setting. Averaged over datasets, LLaGA accuracy changes by at most 0.81 percentage points for every backbone in both

Table 10: Threshold-free activation score comparisons. Entries report NC / LP, averaged over seven datasets. Top/median is the ratio of the highest token score to the median score, calculated within each example and block before averaging. For LLaGA, PAD AUROC measures how often a padding token scores above a real-node token, and target percentile gives the target node's rank percentile, with 1 denoting the highest score.

Backbone	LLaGA top/median	TEA-GLM top/median	LLaGA PAD AUROC	LLaGA target percentile
Vicuna-7B	4.35 / 4.85	1.61 / 1.62	0.92 / 0.90	0.15 / 0.23
Qwen2.5-7B	2.85 / 3.32	1.61 / 1.61	0.85 / 0.86	0.33 / 0.36
Qwen2.5-14B	1.83 / 2.39	1.19 / 1.18	0.91 / 0.91	0.20 / 0.23
Qwen2.5-72B	2.51 / 3.53	1.15 / 1.15	0.87 / 0.86	0.17 / 0.24
Llama-3.1-8B	2.49 / 3.89	1.24 / 1.27	0.98 / 0.97	0.12 / 0.25
Llama-3.1-70B	2.34 / 3.58	1.24 / 1.23	1.00 / 0.99	0.29 / 0.33
Yi-1.5-9B	1.75 / 2.00	1.26 / 1.27	1.00 / 1.00	0.14 / 0.25

tasks. TEA-GLM shows changes of at most 0.81 points for six backbones, while Vicuna-7B accuracy decreases by 5.86 points in NC and 3.57 points in LP. The small effects of top-2 exclusion in LLaGA therefore remain when tokens are selected by rank alone. For TEA-GLM, the larger decreases on Vicuna-7B show that this result varies across backbones. These experiments do not include matched random-token controls.

Table 11: Accuracy changes (%) after excluding the two highest-scoring graph tokens at every block without a token threshold. Results are averaged equally over seven datasets, using 300 examples per setting. LP averages include settings with chance-level baseline performance. This intervention excludes tokens rather than cumulatively deleting them from the sequence.

	LLaGA		TEA-GLM	
Backbone	NC	LP	NC	LP
Vicuna-7B	+0.19	-0.10	-5.86	-3.57
Qwen2.5-7B	-0.33	+0.81	-0.19	-0.19
Qwen2.5-14B	+0.10	+0.14	-0.24	-0.52
Qwen2.5-72B	-0.10	+0.05	+0.33	-0.38
Llama-3.1-8B	-0.10	-0.10	+0.19	+0.05
Llama-3.1-70B	+0.14	-0.24	+0.38	-0.81
Yi-1.5-9B	-0.48	-0.76	-0.19	-0.19

For the ten completed LLaGA sweep settings, we vary the number of selected tokens over $k \in \{1, 2, 4, 8, 16\}$ and compare top-k exclusion with random-token and random-padding controls (Table 12). Top-k exclusion produces changes between -2.00 and $+2.00$ percentage points, with no consistent decline as k increases. At $k = 16$, Vicuna-7B NC decreases by 2.00 points, whereas Qwen2.5-7B NC increases by 2.00 points. Together with the detection ranks, these results support the distinction between activation magnitude and predictive contribution beyond the choice of $\tau_{\text{tok}} = 20$.

Table 12: Rank-based exclusion with matched controls in the ten completed LLaGA sweep settings. Entries are accuracy changes in percentage points relative to the paired unmodified baseline, using 150 examples per setting. Tokens are selected at every block without τ_{tok}. Random controls use two seeds. The LP average includes four settings with chance-level baselines.

Selection budget k	1	2	4	8	16
Vicuna-7B, NC					
Top-k	+0.00	+0.00	+0.00	+0.00	-2.00
Random tokens	-1.00	-0.67	-1.00	-1.67	-0.67
Random `[PAD]`	+0.00	+0.00	+0.00	+0.00	-0.33
Qwen2.5-7B, NC					
Top-k	+1.00	+0.00	+0.33	+1.67	+2.00
Random tokens	+0.83	+0.33	+0.50	+1.67	+0.17
Random `[PAD]`	-0.17	+0.50	+0.67	+1.33	+1.17
Vicuna-7B, LP					
Top-k	+0.19	-0.10	-0.10	-0.86	+0.38
Random tokens	-0.24	-0.29	-0.38	-0.05	-0.62
Random `[PAD]`	-0.14	-0.14	-0.38	-0.48	-0.71

C Limitations and Future Work

In this study, we focus on two representative GLM architectures, LLaGA and TEA-GLM, and on node classification and link prediction tasks. For TEA-GLM, we follow the original authors' setup and evaluate zero-shot transfer to target datasets without task-specific fine-tuning. Although explicitly noted in the main text, we note that performance in this setting can vary across datasets, which should be considered when interpreting the effects of graph-token interventions. Comparing zero-shot transfer alongside the fine-tuned supervised settings would help clarify how training affects the relationship between sink activations and predictive contribution. More broadly, future work could extend our analysis to additional GLM architectures, and examine whether changes to graph token construction, placement, and alignment improve the use of graph information. Studying how sink behavior emerges fundamentally for graph modalities would also help to propose certain mitigation techniques or rather better utilization of these tokens for graph tasks.

D Implementation Details

Backbones and checkpoints. We evaluate LLaGA and TEA-GLM with seven LLM backbones: Vicuna-7B-v1.5, Qwen2.5-7B/14B/72B-Instruct, Llama-3.1-8B/70B-Instruct, and Yi-1.5-9B-Chat. The LLM parameters remain frozen throughout training; only the graph-side modules are updated. We use seed 42 for training.

LLaGA. For Vicuna on the citation datasets, we use the publicly released LLaGA checkpoint with `vicuna-7b-v1.5-16k`. We train the projectors for the remaining settings following the LLaGA setup. The projector is a two-layer MLP with GeLU activation. We use the Neighborhood Detail (ND) template with two hops and a sampling of 10 at each hop, producing 111 graph tokens per node. The missing neighbors are represented by `[PAD]` tokens. Node features are obtained from SimTeG (Duan et al., 2023) for citation datasets and SBERT (`all-MiniLM-L6-v2`) for e-commerce datasets, and concatenated with a 111-dimensional Laplacian positional encoding. Projectors are trained for one epoch using AdamW with learning rate 2×10^{-3}, no weight decay, a cosine schedule, warmup ratio 0.03, effective batch size 16, and maximum sequence length 2048. Training uses bfloat16. Prompts follow the Vicuna-style `USER`/`ASSISTANT` format with each backbone's own end-of-turn token.

TEA-GLM. TEA-GLM uses a GraphSAGE encoder and a projector that produces five graph-token embeddings. Following the original authors' transfer setup, we train citation checkpoints on Arxiv and e-commerce checkpoints on Computers, then evaluate the other datasets in each domain zero-shot. All TEA-GLM checkpoints use NC training instructions, so LP is also evaluated zero-shot.

For non-Vicuna backbones, training has two stages. First, we train a two-layer GraphSAGE encoder with mean aggregation, hidden size 2048, and output size equal to the LLM hidden size. The contrastive alignment uses the first 1000 principal components of the backbone's token-embedding matrix. This stage uses Adam with learning rate 2×10^{-3}, batches of 512 nodes, sampling fan-out $(25, 10)$, and early stopping based on training loss. Second, we freeze the encoder and train a linear projector followed by LayerNorm on $20{,}000$ NC instructions for three epochs. Projector training uses AdamW with learning rate 10^{-4}, no weight decay, a cosine schedule, warmup ratio 0.03, effective batch size 32, maximum text length 700, and bfloat16. We evaluate the final checkpoint. The Vicuna citation checkpoint uses a separate configuration: its projector is trained on Arxiv for 50 epochs with batch size 4, learning rate 10^{-3}, weight decay 0.1, and the same schedule and warmup ratio.