GraphGhost: Tracing Structures Behind Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) demonstrate remarkable reasoning capabilities, yet the structural mechanisms underlying these abilities remain under explored. In this work, we introduce GraphGhost, a unified framework that represents neuron activations and their signal propagation as graphs, explaining how LLMs capture structural semantics from sequential inputs and generate outputs through structurally consistent mechanisms. This graph-based perspective enables us to employ graph algorithms such as PageRank to characterize the properties of LLMs, revealing both shared and model-specific reasoning behaviors across diverse datasets. We further identify the activated neurons within GraphGhost and evaluate them through structural interventions, showing that edits to key neuron nodes can trigger reasoning collapse, altering both logical flow and semantic understanding. Together, these contributions position GraphGhost as a powerful tool for analyzing, intervening in, and ultimately understanding the structural foundations of reasoning in LLMs.

1 Introduction

Large Language Models (LLMs) have demonstrated promising capabilities in capturing structural information from text (Sui et al., 2024; Wang et al., 2025, 2023). Notable examples include identifying relationships between entities in context (Zhang and Soh, 2024), reasoning and planning through structured thinking processes (Yao et al., 2023a; Besta et al., 2024; Wu et al., 2024), and recognizing logical connections across paragraphs (Pan et al., 2023). Trained on large-scale corpora, recent studies suggest that LLMs can transfer reasoning processes by extracting semantics and generalizing learned reasoning strategies (Muennighoff et al., 2025). This raises the question of whether LLMs internally construct reasoning structures and leverage them across diverse tasks.

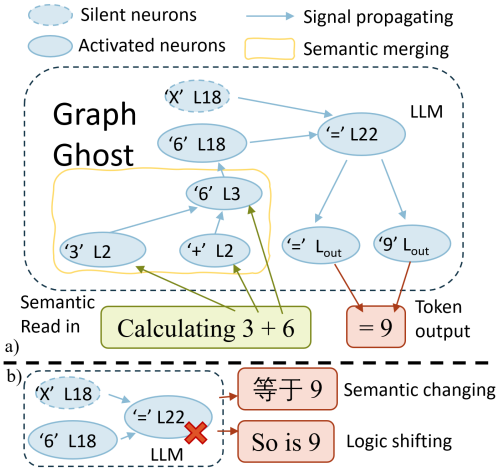


Figure 1: An illustration of GraphGhost based on Qwen3-0.6B. a) GraphGhost illustrates a structural mechanism within LLMs that operates consistently across datasets. L indicates the layer number. Sequential inputs activate subsets of neurons, which merge tokens in the intermediate layers to capture semantic meaning. These activated neurons then propagate signals to the logit layers to generate outputs. b) Muting the neuron node. Changing the structure of GraphGhost leads to corresponding changes in both language expression and logical reasoning. "等于" means "equal to" in Chinese.

To understand the mechanisms of constructing reasoning structures from text, prior studies have often simplified reasoning tasks into toy examples such as graph-based reasoning to examine how LLMs trace reasoning paths within giving structures (Sanford et al., 2024). The underlying hypothesis is that uncovering how decoder-only transformers construct such graphs may yield insights that generalize to real-world tasks (Saparov et al., 2025). While these toy graph settings provide preliminary insights—highlighting mechanisms such as token merging (Dai et al., 2025b) and edge memorization (Wang et al., 2024), they fall short when applied to complex, real-world structured data. To bridge this gap, we introduce GraphGhost, a framework that provides a structural perspective on how LLMs across different architectures perform reasoning tasks in diverse domains. GraphGhost cap-

tures neurons and their signal propagation inside LLMs, revealing how neurons are activated across samples and datasets, enabling shared semantic understanding and supporting the construction of reasoning processes across contexts.

As illustrated in Figure 1 a), GraphGhost represents a structural mechanism modeled within LLMs, where neuron activations capture semantic information and guide token outputs. In the sentence-reading stage, LLMs merge tokens to derive semantic meaning, thereby activating specific neurons associated with those tokens. These activated neurons then propagate signals to higher layers, ultimately influencing the logit neurons in the output layers. Notably, while certain neurons are activated for the given samples, many remain silent. These silent neurons are reserved for unseen inputs from different domains, enabling shared output behaviors across contexts. For example, the symbol 'X' may activate the '=' symbol in mathematical equations, even if the current sample does not require it, leaving those neurons in a silent state until needed.

We define neurons as nodes and the signal propagation among neurons as edges. This representation allows us to analyze LLMs from a graph perspective. For instance, by inspecting high in-degree nodes or applying PageRank, we can identify key nodes that control next-token prediction, as further elaborated in Section 3. Furthermore, we evaluate the effectiveness of these nodes through structural interventions, such as node removal (Figure 1 b)). Removing specific nodes can lead to semantic shifts, such as switching to another language, or logical changes. In addition, we conduct statistical analyses of these structural perturbations to quantify their effects, as presented in Section 4. In conclusion, our key contributions are summarized as:

1. We introduce GraphGhost, a unified framework represented in graphs, explaining how LLMs capture structural semantics from sequential inputs and generate outputs through a structurally consistent mechanism.

2. We provide a graph-based perspective for understanding LLMs and apply graph algorithms on GraphGhost to characterize the properties of LLMs, revealing both unique and shared behaviors across different models and datasets.

3. We identify the activated neurons within LLMs and evaluate them by editing the structure of GraphGhost. Our experiments show that key neuron nodes can trigger reasoning collapse, altering both the logical flow and semantic understanding.

2 Related work and Preliminary

GraphGhost is built on circuit tracing, which is a typical LLM interpretation tool. In this section, we first introduce the background on LLM interpretability and reasoning tasks in Section 2.1 and Section 2.2, respectively. To illustrate how LLMs construct internal structures that align with explicit reasoning texts, we design a toy experiment in Section 2.3 to illustrate the alignment of inner signal propagation with explicit structure descriptions. Building on this foundation, we provide the preliminary definition in Section 2.4.

2.1 Interpretation of LLMs

Interpreting how LLMs perform reasoning is essential for aligning their behavior with human understanding (Zhao et al., 2024). Existing approaches fall into two categories, self-explanation (Gu et al., 2024) and causal-mechanistic interpretation (Bloom et al., 2024). The details are suggested in Appendix A.1

The causal-mechanism approach focuses on explaining how each token is predicted (Anthropic, 2023; Ameisen et al., 2025). Based on learning the internal reasoning pathways, Lindsey et al. (2025) introduced attribution graphs to capture token flows that contribute to final predictions. Formally, an attribution graph is defined as $G_A = (\hat{V}, \hat{E}, A)$, where $\hat{V}$ denotes activated neurons, $A_{s \to t}, ; s, t \in \hat{V}$ represents the contribution from node s to t (computed via backward Jacobians), and $\hat{E}$ indicates whether an edge exists. The edge set is controlled by an edge ratio, which filters the top influential connections according to A.

2.2 Reasoning Tasks

Reasoning tasks including mathmatical (Cobbe et al., 2021), scientific (Khot et al., 2020), and logic reasoning (Clark et al., 2019) are set to evaluate the ability in solving complex problems. The details are in Appendix C.2

Graph reasoning represents a distinct type of reasoning task (Dai et al., 2025a). In this setting, models are provided with textual descriptions of graph structures and asked to reason over them. Success on such tasks suggests that LLMs can

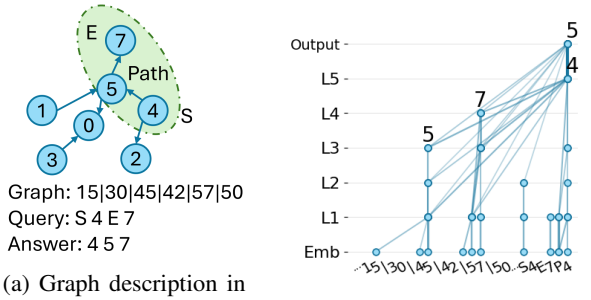
leverage implicit structural representations to guide reasoning (Wang et al., 2023). However, how these implicit structures are internally constructed and maintained within LLMs remains an open question.

2.3 Path tracing in the decoder-only transformer

Graph reasoning tasks, particularly path reasoning, offer a prototypical setting for understanding how structural information in sequential inputs can be mapped to outputs that describe those structures (Wang et al., 2023). In this task, the model is provided with a graph description and a query consisting of a start node and an end node, and the objective for LLMs or decoder-only transformers is to predict the path connecting the two nodes (Dai et al., 2024; Wang et al., 2024). We adopt this basic setup to investigate how decoder-only transformers perform path reasoning. Following prior work (Wang et al., 2024), we use simplified graph descriptions, and a query ("from start point S to end point E") and must predict the connecting path. as illustrated in Figure 2a, and train a 5-layer GPT-2 architecture decoder-only transformer on this task. The trained model achieves 98% accuracy on the test set. We then apply the circuit-tracing method, specifically using transcoder (Dunefsky et al., 2024), to analyze how the transformer models the path-finding process, as shown in Figure 2b. Full experimental details are provided in Appendix C.1.

Circuit tracing reveals how decoder-only transformers review the previous context to identify related nodes and predict the next token. From a structural understanding perspective, neighboring nodes are merged in the middle layers. For example, at the third layer, node 4 is merged with node 5, forming the semantic representation of the edge between them. Further, in generating the output tokens, the final token in the sequence receives the message that an edge exists between 4 and 5 at layer 3, leading the model to predict 5 as the next token.

From this toy example, we observe that decoder-only transformers are not only capable of reasoning about paths within given graph structures, but that their underlying mechanism involves activated neurons propagating signals across layers. However, beyond the toy example with decoder-only transformers, does this mechanism extend to LLMs that can solve more complex tasks expressed in natural language? To address this, we introduce GraphGhost, a unified framework that represents these neuron activations and their signal flows as graphs. GraphGhost enables us to trace and mergeattribution graphs defined in Section 2.1 across diverse reasoning tasks, revealing how LLMs encode and reuse structural reasoning patterns. In the following sections, we apply GraphGhost to multiple models and domains to analyze how their internal reasoning structures build up, interact, and generalize.



(a) Graph description in a simplified way. The sequence includes the graph description, query, and answer.

(b) Interpretation of the graph path reasoning task. The bolder lines indicate stronger contributions to neuron activations.

Figure 2: A toy example of path reasoning on a graph. (a) The simplified descriptions to predict the path between 4 and 7; (b) The circuit tracing for the interpretation of why token 5 is selected to be predicted.

2.4 Preliminary definitions

We formalize the core concepts underlying GraphGhost. GraphGhost is constructed from attribution graphs obtained using CircuitTracer$(\cdot)$, which traces neuron activations contributing to next-token prediction and records their signal flow. Let $X = (t_1, \ldots, t_n)$ be the input tokens of an LLM with layers $\{f^{(1)}, \ldots, f^{(L)}\}$. For a target token t_{n+1}, the attribution graph is defined as $A_{n+1} = \text{CircuitTracer}(t_{n+1})$, where each node represents an activated neuron (t_i, ℓ) with activation $h^{(\ell)}(t_i) = f^{(\ell)}(h^{(\ell-1)}(t_i))$, and each directed edge $w_{s\to t}$ encodes the contribution from neuron s to t. The set of contributing tokens is $\hat{T}(t_{n+1}) = \{\, t_i \mid (t_i, \ell) \in A_{n+1} \,\}$. where ℓ denotes the layer index. These local attribution graphs form the building blocks of GraphGhost, which aggregates them across samples into a unified weighted structure.

Following the definition of attribution graphs in Ameisen et al. (2025), a *neuron node* is represented as a pair (t_i, ℓ), where t_i denotes a token and ℓ denotes the layer index. Each neuron node has a corresponding hidden activation $h^{(\ell)}(t_i)$, representing the hidden state of token t_i at layer ℓ, computed

as $h^{(\ell)}(t_i) = f^{(\ell)}\big(h^{(\ell-1)}(t_i)\big)$. For a source neuron node $s = (t_i, \ell_s)$ and a target neuron node $t = (t_j, \ell_t)$ with $\ell_s < \ell_t$, the edge weight in the attribution graph is defined as $w_{s\to t} = A_{s\to t}$, where $A_{s\to t}$ denotes the attribution score from s to t.

For example, in Figure 2,to generate token 5, the contributing tokens are $\{4, 5, 7, \texttt{"|"}, \texttt{"S"}, \texttt{"P"}\}$. The corresponding neuron nodes include $h^{(3)}(5)$, $h^{(4)}(7)$, and $h^{(5)}(4)$, among others. The edge weight from $h^{(3)}(5)$ to $h^{(5)}(4)$ is denoted as $w_{h^{(3)}(5)\to h^{(5)}(4)}$.

3 GraphGhost in LLMs

Beyond the toy examples, we apply *GraphGhost* to real-world LLMs to uncover their internal reasoning structures. GraphGhost traces neuron activations that contribute to next-token generation, represents them as local attribution graphs, and merges these across samples into a global weighted graph. This unified structure reveals stable signal propagation patterns shared across tasks and models. Section 3.1 details the tracing and aggregation process. To evaluate whether these structures correspond to meaningful reasoning behaviors, we introduce the experimental setup in Section 3.2. which specifies the models, datasets, and tracing configurations used for constructing GraphGhost. Section 3.2 and Section 3.4 then present case studies and quantitative analyses based on these settings.

3.1 Tracing the GraphGhost

To construct a unified GraphGhost that captures reasoning structures shared across multiple samples, we aggregate the attribution graphs obtained from each sample. Given a set of samples $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$, where each sample s_i contains an answer token $t_n^{(i)}$, we initialize a global graph $G = (V, E, W)$ with empty node, edge, and weight sets. For each sample s_i, we start from its answer token $t_n^{(i)}$ and iteratively trace the contributing tokens and their local attribution graph using the CircuitTracer$(\cdot)$ function. Each traced graph produces a set of neuron nodes $\hat{V}$ and directed edges $\hat{E}$, which are merged into the global GraphGhost. If an edge $e \in \hat{E}$ does not exist in E, it is added with an initial weight $W(e) = 1$; otherwise, its weight is incremented by one. This cumulative weighting reflects the frequency of edge occurrences across samples, capturing the stability and recurrence of signal propagation pathways. After processing all samples, the resulting GraphGhost $G = (V, E, W)$ represents a multi-sample structural abstraction of the model's internal reasoning dynamics, which is a weighted directed graph describing neuron-level signal propagation. The full procedure is summarized in Algorithm 1 fromAppendix B.

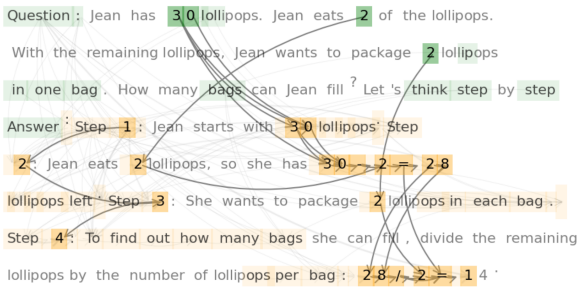


Figure 3: Case study of tracing contributing tokens for the answer 14 in Qwen3-0.6B model. We highlight the numerical reasoning flow captured by GraphGhost to visualize how number-related tokens contribute to the final prediction.

3.2 Experiments setting

We evaluate a range of LLMs and datasets to investigate whether GraphGhost constitutes a general mechanism across models. Specifically, we consider Qwen3-0.6B (28 layers), Qwen2.5-1.5B (28 layers), Llama3.1-8B-Instruct (32 layers), DeepSeek-distilled-Qwen2.5B-1.5B (DS-Qwen-1.5B) (28 layers), and DeepSeek-distilled-Llama-8B (DS-Llama-8B) (32 layers). For datasets, we examine three domains: (i) mathematical reasoning (GSM8K, MAWPS), (ii) logical reasoning (ProntoQA, BoolQA), and (iii) scientific reasoning (ARC-Easy, QASC). To elicit the models' reasoning processes, we apply Chain-of-Thought prompting. Following (Hanna et al., 2025), we then train transcoders at every layer for each dataset–model pair to probe neuron activations during reasoning, ensuring that the transcoders effectively capture the most strongly activated neurons. All the experiments are run on a single H200 GPU. Full experimental details are provided in Appendix C.2.

3.3 Case study on GraphGhost

We further provide some case studies of GraphGhost on contributing nodes and substructures, respectively. Figure 3 is an illustrative example of tracing contributing tokens within a single sample. The identified token set T includes tokens from both the input question (highlighted in green) and those generated along the reasoning chain (in orange). As shown in the figure, number-related tokens emerge through three distinct mechanisms:

Figure 4: A subgraph of GraphGhost in Qwen3-0.6B model. We include 4 datasets from different domains.

(i) step-by-step progression (e.g., 1–2–3–4), (ii) direct copying from the input context (e.g., 30 and 28), and (iii) generation influenced by neighboring tokens, such as the digit "1" in the final answer "14", which is activated by "=". This tracing process demonstrates how local dependencies between tokens collectively form reasoning paths during sequential generation.

Figure 4 demonstrates a representative subgraph extracted from the GraphGhost constructed on Qwen3-0.6B model, integrating neuron activations from mathematical reasoning (GSM8K and MAWPS), scientific reasoning (ARC), and logical reasoning (ProntoQA). Each neuron node is annotated with its corresponding token and layer ID. For example, the activated period at layer 22 is denoted as $._{22}$. This figure highlights how neuron activations from diverse domains converge into a unified structural representation, revealing consistent signal propagation patterns underlying different types of reasoning tasks.

In the subgraph, we observe that some nodes are activated consistently across datasets, while others contribute only to specific domains or even individual samples. For example, the period symbol "." at layer 22 triggers strong logit activations across all four datasets, and the token "So" frequently appears across different reasoning types. Domain-specific reasoning tokens also emerge within the same domain. For instance, the action "Adding" and arithmetic expressions such as "6+" appear in both GSM8K and MAWPS. In contrast, certain nodes are unique to individual samples—for example, B_{18} activates only in choice-option reasoning within the ARC dataset.

3.4 Analysis on GraphGhost

In Figure 4, we also observe two typical actions associated with the node $._{22}$. The first is semantic understanding, exemplified by the "6+" operation. The second is logic shifting, where the activation leads to the token "So" in the subsequent prediction. We further analyze these two actions from a statistical perspective.

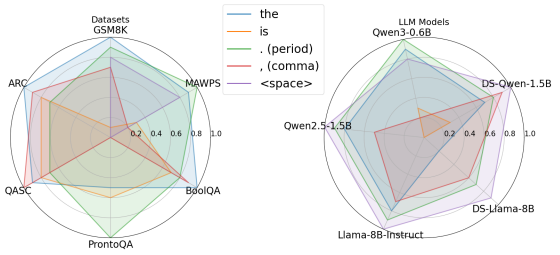


Figure 5: The top merging nodes across the models and datasets.

Semantic understanding In Section 2.3, we discussed how edge information is merged within the inner layers. Building on this, we analyze the token-merging behavior of neuron nodes by examining their average in-degree across layers in LLMs' settings. The statistical results, summarized in the Appendix D and illustrated in Figure 5, show the top tokens across different datasets and models. We observe that the tokens with the highest in-degree are domain-sensitive. For example, in the ProntoQA dataset, the model is more sensitive to the period token than to the comma. This suggests that LLMs activate specific neurons in response to domain-specific data to capture semantics. Across models, however, the activated neurons are generally consistent: tokens such as <space>, period, and "the" consistently appear among the top-ranked, and this alignment persists even after finetuning.

We further analyze the merging behavior across layers by computing the *in-degree ratio*. Let $d_i^{(\ell)}$ denote the in-degree of neuron node (t_i, ℓ), where ℓ is the layer index and i indexes the neuron within that layer. The in-degree ratio of layer ℓ is then defined as $R_\ell = \frac{\sum_{i=1}^{N_\ell} d_i^{(\ell)}}{\sum_{k=1}^{L}\sum_{i=1}^{N_k} d_i^{(k)}}$, where N_ℓ is the number of neurons in layer ℓ and L is the total number of layers. As shown in Figure 6, the in-degree ratio distribution is imbalanced across both models and tokens. Although Qwen models and the Llama model perform semantic merging primarily in the bottom and top layers, the exact layers differ between them. Similarly, token merging patterns are not always consistent. Most tokens are merged in shallow layers, whereas high in-degree tokens such as period and "the" tend to merge at higher layers.

Logic shifting. The reasoning structure is segmented by special tokens such as So, Step, and Wait (Muennighoff et al., 2025; Bogdan et al., 2025). We assume that activated neuron nodes in the middle layers trigger these special token expressions at the logit layers. At the same time, the neuron nodes themselves are activated by input to-

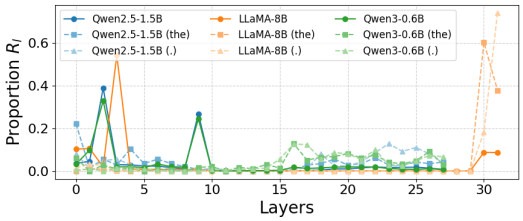


Figure 6: The in-degree proportions on various layers with MAWPS dataset

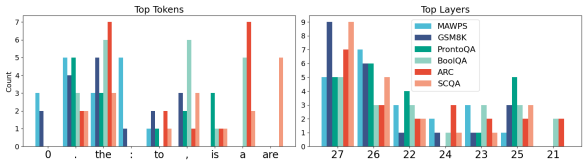


Figure 7: Top contributing tokens and layers in Qwen3-0.6B in various datasets.

kens. To capture both effects, we treat logit-layer nodes as signals directed back to the neuron nodes by reversing their edge directions. We then apply the PageRank algorithm to identify the most influential nodes. We summarize the results in Figure 7 and Figure 8, and the full details can be found in Appendix D.

Figure 7 presents the token distribution across datasets, highlighting the distinction between silent and activated tokens in LLMs when processing data from different domains. For example, numerical tokens such as 0 are more frequently activated in mathematical datasets, while general tokens like the period and “the" appear across multiple domains. Furthermore, Figure 8 illustrates that different models exhibit distinct activation preferences. After finetuning with distilled data, DS-Qwen-1.5B and DS-Llama-8B show increased activation of the token "**", suggesting that DeepSeek-distilled data encourages reliance on this token during generation. From a layer-level perspective, activated nodes consistently emerge in higher layers across both models and datasets, though not exclusively in the final layer; in the Qwen series, layers 22 to 27 frequently contain critical activated nodes.

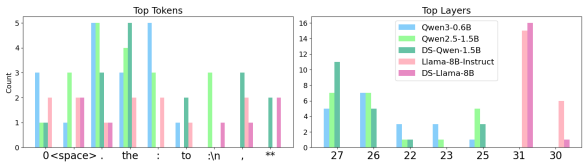


Figure 8: Top contributing tokens and layers across models evaluated by MAWPS dataset.

4 GraphGhost affects the reasoning process

GraphGhost paves a new way in which LLM reasoning processes can be analyzed and controlled. To evaluate this, we perform perturbation experiments at two complementary levels. (i) At the input level, we modify the contributing tokens identified in Section 4.1 to examine how altering key linguistic elements affects the model's reasoning trajectory and output behaviors. (ii) At the neuron level, we intervene on the influential neuron nodes detected via in-degree analysis and PageRank in Section 4.2, assessing how structural disruptions within the reasoning pathways influence the model's internal logic flow. Together, these two perturbation perspectives reveal how surface-level token changes and internal structural shifts jointly shape the reasoning dynamics captured by GraphGhost.

4.1 Impact of contribution tokens

GraphGhost in LLMs leverages contributing tokens to capture context and perform reasoning. In practice, we assess their importance by masking tokens, replacing each with a blank token. To quantify this effect, we evaluate two metrics: Sparsity and Fidelity. Sparsity measures the proportion of contributing tokens in a given question, while Fidelity evaluates whether the answer generated from only the contributing tokens remains consistent with the original answer. Formally, $\text{Sparsity} = \frac{\#C}{\#T}$ and $\text{Fidelity}_T = \mathbb{1}(LLM(C), LLM(T))$, where C denotes the set of contributing tokens, T is the full set of tokens, and $\mathbb{1}$ is an indicator function returning 1 if the two answers are identical and 0 otherwise. In practice, we use gemini-2.5-flash-lite to identify whether the two predictions are the same, with the prompt as: "Comparing two answers for a question: Answer1 $\{LLM(\hat{T})\}$, Answer2 $\{LLM(\hat{T})\}$. Do they have the same final results?" To compare the effectiveness of GraphGhost across models, we evaluate 100 samples that are solvable by Qwen3-0.6B, Qwen2.5-1.5B, and DS-Qwen-1.5B. The results are presented in Figure 9.

As shown in Figure 9, compared with random masking of the given question, the contributing tokens identified by GraphGhost successfully preserve the generated answer. This effect is particularly significant in mathematical reasoning tasks such as MAWPS and GSM8K, where even when half of the question tokens are removed, LLMs

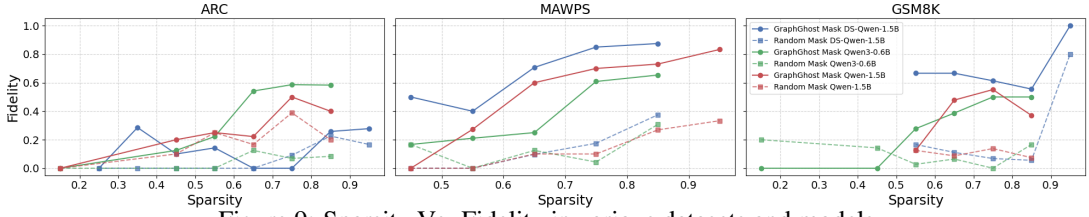


Figure 9: Sparsity Vs. Fidelity in various datasets and models.

Change the semantic

Change the language
Question: A) nervous and digestive B) nervous and circulatory C) respiratory and digestive D) respiratory and circulatory Question: Which two body systems would most directly remove extra fluid from a person's lungs?
Original answer: ... B) Nervous and circulatory systems: The nervous system is responsible for regulating the body's response to stress, while the circulatory system is responsible for transporting blood throughout the body. Neither of these systems directly remove extra fluid from the lungs...
Mute the neuron node 'the' at layer 20
Perturbed answer: ... B) Nervous and circulatory systems: The nervous system is responsible for controlling the - circulatory system is responsible for transporting blood throughout the分区肺. While the分区肺需要血液来清除多余的水分，这些系统本身并不直接参与这个过程。...

Change the word expression
Question: To determine how many trips Jerry will make, we need to calculate the total number of trays he needs to pick up and then divide by the number of trays he can carry at a time.
Original answer: ... we divide the total number of trays, 16.0 trays ÷ 8.0 trays per trip = 2 trips...
Mute the neuron node '.' at layer 17
Perturbed answer: ... Number of trips = Total trays / 8 = 16 / 8 = 2 trips ...

Figure 10: Changing the semantic understanding. The characters in changing the language case is translating the original answer.

are still able to produce the correct prediction. In contrast, the performance gap between randomly masked and selected tokens is smaller in scientific reasoning tasks such as ARC dataset. This is because in ARC the answers often involve selecting the correct option, and in some cases LLMs rely on direct recognition of the options from the given context rather than explicit reasoning.

4.2 Modification on the graph structure

To evaluate whether GraphGhost successfully identifies the most influential tokens in LLMs, we mute specific neuron nodes and observe the resulting changes in the outputs. Specifically, neuron nodes are muted by setting the corresponding positions of their layer–token pairs to zero during the reasoning process. We evaluate the impact through both case studies and statistical analysis.

Case study First, as discussed in Section 3.4, the impacts include changes in semantic understanding and logic shifting. As shown in Figure 10, the semantic understanding is reflected in changing the language formulation and word expression. When the neuron node "the" is deleted at layer 20, the English semantic pathway is disrupted, activating Chinese neurons that share the same semantic signal. As a result, the output appears in Chinese, directly translated from English. Furthermore, in the word-expression change case, deleting the neuron "." at layer 17 causes the token "/" to replace "÷." This suggests that another neuron is responsible for propagating the division signal, expressing it as "/." These findings indicate that LLMs store semantics redundantly across multiple neuron nodes: when the most relevant neuron is muted, the model relies on alternative neurons that produce similar expressions. In Figure 11, we demonstrate how GraphGhost influences reasoning by altering key steps or driving the generation into loops. In the key-step change case, deleting the neuron node "." at layer 25 introduces a new reasoning token such as "So", ultimately leading the model to produce the correct answer. In the loop case, deleting the neuron node "the" at layer 27 causes the reasoning to degenerate into a repetition, producing "theShaper Information." These cases suggest that even a single neuron in the middle layer can fundamentally change the outcome of the reasoning process.

Statistical analysis To evaluate whether the discovered nodes have a general impact across different data predictions, we perturb the model structure according to the neuron nodes discussed in Section 3. We then report $\text{InFidelity} = 1 - \text{Fidelity} = \mathbb{1}(LLM(T), L\hat{L}M(T))$, which indicates cases where muting the nodes in $L\hat{L}M(\cdot)$ completely changes the model's prediction. Although GraphGhost is constructed using only 100 data samples, we demonstrate that the perturbations influence the entire dataset, including the 1,772

Change the reasoning structure

Change the key steps

Question: There are 7.0 crayons in the drawer and 6.0 crayons on the desk. Sam placed 4.0 crayons and 8.0 scissors on the desk. How many crayons are now there in total? Let's think step by step (ground truth: 17)

Original answer: ... - Sam placed 4.0 crayons on the desk. - Sam also placed 8.0 scissors on the desk. .... - Total crayons on the desk = 7.0 + 4.0 + 8.0 - Total crayons on the desk = 19.0

Mute the neuron node '.' at layer 25

Perturbed answer: ... - Sam placed 4.0 crayons on the desk. - So, the total number of crayons on the desk is now 6.0 (initial crayons) + 4.0 (Sam's crayons) = 10.0 crayons. ... - So, the total number of crayons = 7.0 (drawer) + 10.0 (desk) = 17.0 crayons.

Failed into loop

Question: Alyssa picked 25.0 limes and Mike picked 32.0 limes. Tom picked 12.0 plums. How many limes were picked in all? Let's think step by step

Original answer: Step 1: Identify the number of limes picked by Alyssa and Mike. Alyssa picked 25.0 limes. Mike picked 32.0 limes. ...

Mute the neuron node 'the' at layer 27

Perturbed answer: Step 1: Identify theSharper Information In this problem, we are given theSharper Information about theSharper Information about Alyssa, Mike, and Tom's limes and plums. We need to find theSharper Information about theSharper Information about theSharper ...

Figure 11: Changing on the logic shifting

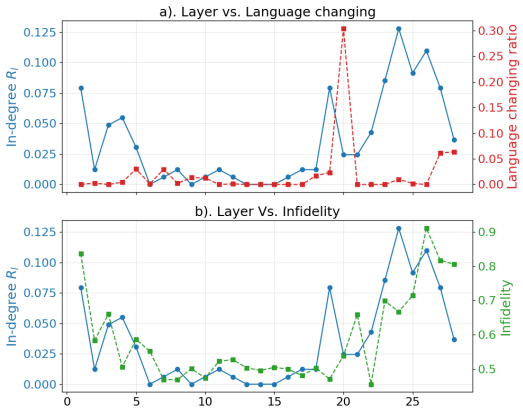


Figure 12: The impact of tokens on various layers. We mute the period token '.' for each layer.

	Top1-10		Top11-20	
	AVG	MAX	AVG	MAX
Token	-	<space>_27	-	0_23
Qwen3-0.6B	0.5845	0.9701	0.3756	0.6958
Token	-	<space>_27	-	<space>_31
Qwen2.5-1.5B	0.4586	0.9577	0.3411	0.8956
Token	-	<space>_31	-	<space>_30
Llama-8B	0.2808	0.8380	0.2454	0.5028

Table 1: The impacts of top tokens. "MAX" denotes the maximum influential token among the top-K slots

samples in MAWPS.

We first evaluate the tokens ranked by the PageRank algorithm. Specifically, we calculate the average InFidelity score when muting the corresponding neuron nodes and identify the maximum-impact neuron nodes within the top-k positions. The results, summarized in Table 1, include the top-1–10 and top-11–20 nodes. Overall, the top-10 nodes show a markedly higher influence on the model's reasoning process, indicating that the PageRank algorithm effectively identifies influential nodes in the GraphGhost framework. Moreover, larger models demonstrate greater stability than smaller ones—for instance, Llama-8B achieves an InFidelity score of 0.28, whereas Qwen3 reaches 0.58. Interestingly, the <space> token emerges as the most impactful token, exerting a strong influence on overall model performance, particularly in higher layers.

We further analyze the relationship between the in-degree ratio across layers and the final reasoning predictions in the Qwen3-0.6B model, as illustrated in Figure 12. Neuron nodes with higher in-degree values are likely to merge semantic information from multiple sources. To verify this, we compute the language-changing ratio when muting individual neuron nodes across layers, defined as $R = \frac{\#\text{non-English answer}}{\#\text{All answer}}$.where a *non-English answer* refers to any generated response that switches from English to another language after perturbation. As shown in Figure 12(a), the peaks of the language-changing ratio align closely with layers exhibiting high in-degree ratios. These layers are primarily distributed in the upper regions of the model, which also correspond to the increase in InFidelity scores observed in Figure 12(b). This pattern suggests that neuron nodes in higher layers are more likely to connect directly to logit neurons, thereby altering the logical flow of reasoning (e.g., causing generation loops) rather than simply substituting equivalent semantic representations.

5 Conclusion

In this work, we introduced GraphGhost, a framework that models neuron activations and their propagation as graphs, providing a structural view of LLM reasoning. Using graph algorithms and structural interventions, we show how key neurons shape the logical flow and semantic consistency. These findings establish GraphGhost as an interpretation framework for understanding the structural foundations of reasoning in LLMs.

Limitations

The fidelity of GraphGhost relies heavily on the precision of the underlying attribution tracing method. Because current attribution estimates can be noisy or incomplete, they may produce missing or spurious edges, yielding only partial views of the model's reasoning process. Advancing the design of circuit tracer would further enhance GraphGhost's ability to identify key neuron nodes and capture more precise reasoning structures.

References

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*.

Anthropic. 2023. Transformerlens transcoder documentation. https://github.com/anthropics/transformerlens.

Fazl Barez, Tung-Yu Wu, Iván Arcuschin, Michael Lan, Vincent Wang, Noah Siegel, Nicolas Collignon, Clement Neo, Isabelle Lee, Alasdair Paren, and 1 others. 2025. Chain-of-thought is not explainability. *Preprint, alphaXiv*, page v1.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.

Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. 2024. Saelens. https://github.com/jbloomAus/SAELens.

Paul C Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. 2025. Thought anchors: Which llm reasoning steps matter? *arXiv preprint arXiv:2506.19143*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 913–923.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Xinnan Dai, Haohao Qu, Yifei Shen, Bohang Zhang, Qihao Wen, Wenqi Fan, Dongsheng Li, Jiliang Tang, and Caihua Shan. 2025a. How do large language models understand graph patterns? a benchmark for graph pattern comprehension. In *The Thirteenth International Conference on Learning Representations*.

Xinnan Dai, Qihao Wen, Yifei Shen, Hongzhi Wen, Dongsheng Li, Jiliang Tang, and Caihua Shan. 2024. Revisiting the graph reasoning ability of large language models: Case studies in translation, connectivity and shortest path. *arXiv preprint arXiv:2408.09529*.

Xinnan Dai, Kai Yang, Jay Revolinsky, Kai Guo, Aoran Wang, Bohang Zhang, and Jiliang Tang. 2025b. From sequence to structure: Uncovering substructure reasoning in transformers. *arXiv preprint arXiv:2507.10435*.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. 2024. Transcoders find interpretable llm feature circuits. *Advances in Neural Information Processing Systems*, 37:24375–24410.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. 2025. circuit-tracer. https://github.com/safety-research/circuit-tracer. The first two authors contributed equally and are listed alphabetically.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2020. QASC: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Kelvin JL Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. 2024. Learning to generate explainable stock predictions using self-reflective large language models. In *Proceedings of the ACM Web Conference 2024*, pages 4304–4315.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. On the biology of a large language model. *Transformer Circuits Thread*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*.

Clayton Sanford, Bahare Fatemi, Ethan Hall, Anton Tsitsulin, Mehran Kazemi, Jonathan Halcrow, Bryan Perozzi, and Vahab Mirrokni. 2024. Understanding transformer reasoning capabilities via graph algorithms. *Advances in Neural Information Processing Systems*, 37:78320–78370.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Abulhair Saparov, Srushti Ajay Pawar, Shreyas Pimpalgaonkar, Nitish Joshi, Richard Yuanzhe Pang, Vishakh Padmakumar, Mehran Kazemi, Najoung Kim, and He He. 2025. Transformers struggle to learn to search. In *ICLR*. OpenReview.net.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

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

Siwei Wang, Yifei Shen, Shi Feng, Haoran Sun, Shang-Hua Teng, and Wei Chen. 2024. Alpine: Unveiling the planning capability of autoregressive learning in language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yuxiang Wang, Xinnan Dai, Wenqi Fan, and Yao Ma. 2025. Exploring graph tasks with pure llms: A comprehensive benchmark and investigation. *arXiv preprint arXiv:2502.18771*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xixi Wu, Yifei Shen, Caihua Shan, Kaitao Song, Siwei Wang, Bohang Zhang, Jiarui Feng, Hong Cheng, Wei Chen, Yun Xiong, and 1 others. 2024. Can graph learning improve planning in llm-based agents? *Advances in Neural Information Processing Systems*, 37:5338–5383.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *ICLR*. OpenReview.net.

Bowen Zhang and Harold Soh. 2024. Extract, define, canonicalize: An llm-based framework for knowledge graph construction. *arXiv preprint arXiv:2404.03868*.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

A Related work

A.1 Interpretation

Interpreting how LLMs perform various reasoning tasks is an important step toward aligning them with human understanding (Yao et al., 2023b; Zhao et al., 2024). Current work mainly falls into two directions: self-explanation (Gu et al., 2024; Koa et al., 2024) and uncovering the underlying causal mechanisms (Bloom et al., 2024; Dunefsky et al., 2024). The self-explanation approach, such as Chain-of-Thought prompting (Wei et al., 2022; Kojima et al., 2022), requires LLMs to verbalize their reasoning steps, which can help guide them to correct answers. However, there is ongoing debate about whether such reasoning chains are truly faithful (Turpin et al., 2023; Barez et al., 2025).

A.2 Reasoning tasks

Reasoning tasks are among the most important benchmarks for evaluating the ability of LLMs to solve complex problems. Specifically, they assess model performance across mathematical, scientific, and logical reasoning. Mathematical reasoning evaluates a model's capacity to perform structured, multi-step numerical computations expressed in natural language (Cobbe et al., 2021; Hendrycks et al., 2021). Scientific reasoning tasks require LLMs to draw on background knowledge and apply it in reasoning processes (Khot et al., 2020; Clark et al., 2018). Logical reasoning tasks assess whether models can apply formal rules rather than relying on surface-level correlations (Clark et al., 2019; Saparov and He, 2022). Recent research further suggests that LLMs are capable of transferring reasoning strategies across diverse tasks, indicating that despite differences in domain, the underlying reasoning logic may be shared.

B Algorithm

Here we present the algorithm of construct the GraphGhost in Algorithm 1.

Algorithm 1 GraphGhost Construction

Input: A set of samples $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$, where each s_i contains an answer token $t_n^{(i)}$
Initialize global GraphGhost $G = (V, E, W)$, $V = \emptyset$, $E = \emptyset$, $W = \emptyset$
for each sample $s_i \in \mathcal{S}$ **do**
  Initialize token list $T = \{t_n^{(i)}\}$
  for each token $t \in T$ **do**
    if Token t not in question of s_i **then**
      # trace contributing tokens and local attribution graph
      $\hat{T}, G'_A \leftarrow \text{CircuitTracer}(t)$
      $T \leftarrow T \cup \hat{T}$
      # update global GraphGhost with local graph
      $\hat{V} = \{v^1(t_1), \ldots, v^\ell(t_n)\}, \hat{E} \leftarrow G'_A$
      $V \leftarrow V \cup \hat{V}$
      for each edge e in $\hat{E}$ **do**
        if $e \notin E$ **then**
          $E \leftarrow E \cup \{e\}$
          $W(e) \leftarrow 1$
        else
          $W(e) \leftarrow W(e) + 1$
        end if
      end for
    end if
  end for
end for
return $G = (V, E, W)$

C Appendix: Experiment setting

C.1 Synthetic Experiment

We first generate directed graphs with 10 nodes and an edge density of 0.6, then randomly drop 60% of the edges. From each resulting subgraph, we randomly select two nodes and retain the subgraph if the path length between them is 3. In total, we collect 100,000 such graph cases. For training, we use a 5-layer decoder-only transformer with a hidden size of 96 and 12 attention heads. The model is trained for 20,000 iterations to ensure stable convergence.

For circuit tracing, we employ the ReLU version of the transcoder, with the dead feature window set to 50 and the L1 coefficient set to 0.0005. During visualization, we set the node ratio to 0.48 and the edge ratio to 0.86.

		Qwen3-0.6B	Qwen2.5-1.5B	Llama-8B	DS-Qwen	DS-Llama
Max Length	MAWPS (1772)	200	200	200	400	400
	GSM8K (1319)	300	300	300	500	500
	ProntoQA (1500)	900	900	900	900	900
	BoolQA (3270)	500	500	500	500	500
	ARC (2376)	300	300	300	500	500
	QASC (926)	300	300	300	500	500
Thinking prompt	Math	Let's...	Let's...	Let's...	Let's... <>	Let's...<>
	Logic	Let's... <>	Let's...	Let's... <>	Let's... <>	Let's... <>
	Science	Let's... <>	Let's...	Let's... <>	Let's...<>	Let's... <>
Transcoder	Edge Ratio	0.4	0.4	0.8	0.4	0.9
	Node Ratio	0.5	0.5	0.9	0.5	0.9
	L1_coefficient	0.0005	0.0005	0.00005	0.0005	0.00005
Training Tokens	MAWPS	5*10^6	5*10^6	5*10^6	5*10^6	5*10^6
	GSM8K	2*10^7	2*10^7	2*10^7	2*10^7	2*10^7
	ProntoQA	1*10^7	1*10^7	1*10^7	1*10^7	1*10^7
	BoolQA	5*10^6	5*10^6	5*10^6	2*10^7	2*10^7
	ARC	2*10^7	2*10^7	2*10^7	2*10^7	2*10^7
	QASC	2*10^7	2*10^7	2*10^7	2*10^7	2*10^7

Table 2: The experiment setting for real world cases. "Let's..." denotes the prompt "Let's think step-by-step", and "<>" denotes "<think>"

Top tokens	Dataset MAWPS	GSM8K	ARC	QASC	ProntoQA	BoolQA
1	.	the	the	,	.	the
2	the	.	,	the	uses	,
3	<space>	<space>	is	is	.\n	.
4	0	,	.	.	ump	is
5	:	2	are	are	is	a
6	1	.\n\n	that	that	the	that
7	.\n	1	us	are	uses	are
8	is	$	.\n\n	of	.\n\n	in
9	,	0	1	to	1	.\n\n
10	.\n\n	is	given	directly	given	was

Table 3: The top tokens sort by the in-degree values across the datasets

C.2 Real-world Experiment

The experimental settings for real-world data are summarized in Table 2. We include both the thinking prompt "Let's think step by step" and the "<think>" tag. For the transcoders, we select parameter configurations that converge to at least an MSE value of 0.001 at the first layer.

D GraphGhost Results

In this section, we present the specific tokens ranked by graph algorithms. For the semantic merging case, where tokens are sorted by in-degree, Table 3 and Table 4 list the top tokens across different datasets and models. Table 7 highlights significant tokens that are merged across multiple layers of LLMs. For logic shifting, we report the top tokens for models and datasets in Table 5 and Table 6, respectively.

Top token	Model Qwen3-0.6B	Qwen2.5-1.5B	DS-Qwen-1.5B	Llama-8B-Instruct	DS-Llama-8B
1	.	<space>	<space>	<space>	contin
2	the	.	,	.	<space>
3	<space>	the	.	the	**
4	0	=	the	,	.
5	:	.\n	:	:	,
6	1	,	.\n\n	=	.\n\n
7	.\n	1	to	to	i
8	is	has	is	0	**
9	,	had	=	of	the
10	.\n\n	\n	0	and	\\

Table 4: The top tokens sort by the in-degree values across the models

Top nodes	Qwen3-0.6B	Qwen2.5-1.5B	DS-Qwen-1.5B	Llama-8B-Instruct	DS-Llama-8B
1	0_27	:\n_27	._27	_31	{_31
2	_27	:_27	the_26	.._31	.\n\n_31
3	._27	:\n_26	the_27	:_31	_31
4	the_26	:_26	,_26	0_31	**_31
5	:_27	_27	,_27	the_31	._31
6	:_26	._27	._26	more_31	.\n_31
7	the_22	0_27	**_27	,_31	**_31
8	._24	.\n_27	the_25	=_31	\n\n_31
9	0_26	_27	,_25	_31	**\n\n_31
10	._26	_25	0_27	the_30	:\n_31
11	of_26	:_25	to_27	,_30	\\n_30
12	0_23	the_26	a_27	of_31	_31
13	._22	._26	._25	:_30	vase_0
14	:_24	_26	the_22	is_31	,_31
15	:_25	the_25	.\n\n_27	0_30	\n_31
16	\n\n_27	._25	**_26	of_30	i_29
17	\n\n_26	the_23	_27	_30	contin_31
18	:_22	_26	the_21	to_31	contin_29
19	._23	:\n_25	I_27	pages_31	contin_30
20	the_23	the_22	to_26	up_31	**\n_31

Table 5: Top tokens ranked by PageRank across LLMs

Top tokens	MAWPS	GSM8K	BoolQA	ProntoQA	SCQA	ARC
1	0_27	the_27	the_27	._26	the_27	the_27
2	_27	0_27	the_26	is_27	._27	a_27
3	._27	._27	,_27	.\n_27	,_27	._27
4	the_26	_27	a_26	._25	,_26	the_26
5	:_27	the_26	a_27	,_25	are_27	a_26
6	:_26	:_27	._27	,_26	is_27	the_22
7	the_22	**_27	the_22	1_27	are_26	the_25
8	._24	of_26	the_25	.\n_22	a_27	a_23
9	0_26	,_27	the_23	ump_27	a_26	,_27
10	._26	._24	,_26	1_26	the_25	the_21
11	of_26	to_26	the_21	._22	)_27	is_27
12	0_23	the_25	is_27	to_25	are_22	the_23
13	._22	,_26	,_22	-_26	._25	a_25
14	:_24	._26	,_21	given_23	the_22	a_22
15	:_25	(_27	._24	the_26	)_26	a_24
16	\n\n_27	the_23	,_20	the_25	"_27	a_21
17	\n\n_26	0_26	a_25	is_26	,_25	to_27
18	:_22	the_22	a_23	_25	to_26	and_27
19	._23	to_25	,_25	ster_27	are_24	._24
20	the_23	,_25	a_22	the_22	of_27	the_24

Table 6: Top tokens ranked by PageRank across datasets

Model	Dataset	Select Token	Layers	# In-degree	New tokens	Description
Qwen3-0.6B	ARC	the	1	26	is, not, of, in, and	Basic function words
			15	141	Because, But, could, bird	Logical connectors/conditionals + noun
			20	289	, Wait, DNA, .\n	Reasoning/pausing words + proper nouns
			25	481	-check, A, 2), Carboon	answer-related symbols and entities
	MAWPS	the	1	None neurons		
			15	40	Calculate, add, in	Math calculation actions
			20	100	1, $, =, Answer, Step	Answer template with math symbols
			25	143	.\n, how, think	Reasoning/reflective tokens
		.	1	2	., :\n\n	Sentence boundary markers
			15	14	Question, Answer, Step	Question–answer framing tokens
			20	134	$, 2, + divide	Mathematical symbols and operations
			25	147	How many, spent, together	Math word-problem phrasing tokens
		,	1	3	First, Next	Sequencing/transition words
			15	1	,	Basic punctuation
			20	11	Since, To	Ttransition words
			25	30	How, total	Question words focusing on totals or amounts
			1	32	Calculate, add, of, in	Function words + common math verbs
Qwen2.5-1.5B	MAWPS	the	15	None neurons		
		.	20	38	Multiply, Question	Math action words
			25	54	1, Answer, step	Answer-step related tokens
			1	16	\n, 1	Formatting symbols and numeric markers
			15	1	.	Basic punctuation
			20	49	=,+, got, Answer, Question	Math operators mixed with answer framing tokens
			25	63	$, *	Mathematical symbols
			1	7	Question, For	Question framing / introductory tokens
Llama-8BI	MAWPS	the	2	18	of, to, ing	Basic function words
		.	30	310	2, \n, If, eggs	Numbers, line breaks, conditional phrasing
			31	313	=, add, or	Mathematical operators and logical connectors
			1	56	1, yen, Question, to, ing	Numeric units and question words
			2	72	all, and, company	Conjunctions and basic nouns
			30	210	2, \n, If, eggs	Numbers with conditionals and real-world nouns
			31	574	=+, 111, dogs	Numbers combined with operators and concrete nouns

Table 7: The examples of how LLMs merge the semantics.