GraphGhost: Tracing Structures Behind Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) exhibit strong reasoning capabilities on structured tasks, yet the internal mechanisms underlying such behaviors remain poorly understood. Existing interpretation methods mainly focus on token-level attributions, which provide limited insight into multi-step reasoning inside the model. We propose GraphGhost, a graph-based framework that models internal token interactions and neuron activations in LLMs as graphs. By aggregating token dependencies traced across layers, GraphGhost captures global information flow underlying model predictions. We formalize GraphGhost from two complementary perspectives: a sample view, which traces token dependencies for individual predictions, and a dataset view, which aggregates recurring structural patterns learned during training. Through graph analytics and quantitative experiments, we show that graph structural properties are closely associated with influential tokens and neuron nodes, and that perturbations to structurally critical nodes lead to measurable changes in reasoning behavior. These results indicate that the structural patterns captured by GraphGhost reflect meaningful internal organization of LLM reasoning. The codes are available at software part. Artifacts will be made available for research use only.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in solving structured reasoning tasks, where solutions require the composition of multiple intermediate steps governed by explicit relational or logical dependencies, such as mathematical problem solving, logical inference, and graph-based reasoning (Bommasani et al., 2022; Pan et al., 2023). Despite this success, the internal mechanisms by which LLMs represent, compose, and reuse such structural information during reasoning remain poorly understood (Nanda et al., 2023; Nananukul et al., 2025; Pandey et al., 2025). This is important because it provides a principled way to understand the source of intelligence in LLMs.

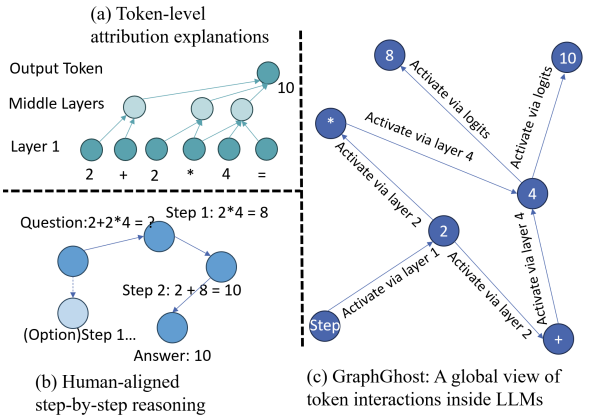


Figure 1: From local explanations and human-aligned reasoning to global internal structures. (a) Token-level attribution explanations provide localized views of individual predictions. (b) Human-aligned step-by-step reasoning produces coherent reasoning traces without revealing internal mechanisms. (c) GraphGhost captures global token interactions inside LLMs as a structured graph, enabling mechanistic analysis of internal reasoning processes.

Most existing efforts to understand LLMs' behaviors on structured reasoning tasks focus on token-level explanations (Chen et al., 2025; Tak et al., 2025), as illustrated in Figure 1(a), which trace individual predictions back to contributing tokens or activations and provide only localized views of isolated outputs. Given that reasoning capability plays a central role in LLM performance across reasoning tasks (Wei et al., 2022; Lewkowycz et al., 2022) as shown in Figure 1 (b), existing token-level approaches provide limited insights into how representations are composed and reused across multiple reasoning steps (Geiger et al., 2025; Dziri et al., 2023). This limitation is increasingly recognized as a major obstacle to meaningful interpretability, as faithful understanding requires alignment between internal reason-

ing mechanisms and human-interpretable reasoning logic (Zhao et al., 2024; Turpin et al., 2023; Barez et al., 2025).

To bridge this gap, we introduce GraphGhost, a graph-based framework that captures internal token interactions and neuron activations in LLMs as graphs as shown in Figure 1(c). Motivated by the central role of step-by-step reasoning in LLM performance, GraphGhost moves beyond isolated token attributions to explore how the reasoning steps are driven under the individual token explanation. This structural representation enables us to analyze reasoning as an internal, mechanistic process, revealing how intermediate tokens are merged, propagated, and reused during generation.

Specifically, in Section 3, we formalize GraphGhost from two complementary perspectives. First, in the sample view, GraphGhost traces internal token dependencies within individual input–output pairs that contribute to a specific prediction, exposing reasoning structures for LLMs. Second, in the dataset view, GraphGhost is constructed by aggregating structures across samples to reveal recurring and shared token-level patterns learned from data. Through graph analytics, we demonstrate that LLM reasoning behavior is tightly coupled to internal graph structural properties, with influential tokens and neuron nodes characterized by structurally critical roles. These findings suggest that internal graph structures fundamentally shape how individual tokens influence LLM reasoning, through their structural roles within the model.

In Section 4, we conduct extensive quantitative experiments to assess whether the structural patterns captured by internal graphs are meaningfully tied to LLM reasoning behavior. Using fidelity, infidelity, and sparsity metrics, we demonstrate that the tokens and neuron nodes identified by GraphGhost exert a significantly strong influence on model behavior. Furthermore, targeted interventions on structurally important nodes lead to measurable changes in reasoning trajectories and outputs, providing empirical evidence that GraphGhost captures meaningful internal structures governing LLM reasoning.

In conclusion, we summarize our key contributions as follows:

1) We propose GraphGhost, a unified framework for understanding how LLMs perform reasoning by modeling internal token interactions and neuron activations at both the sample and dataset levels.

2) Using GraphGhost, we reveal that internal graphs capture structural constraints that govern model behavior, shaping information composition and propagation during reasoning.

3) We conduct quantitative experiments to test whether graphs constructed by GraphGhost capture meaningful structural patterns of LLM reasoning. Our results show that structurally important token-related neuron nodes strongly influence reasoning trajectories, and that intervening on these nodes alters both language expression and logical flow.

2 Related Work

While Chain-of-Thought (CoT) prompting enhances LLM reasoning (Wei et al., 2022; Kojima et al., 2022), its faithfulness to the model's intrinsic computations remains contested (Jacovi and Goldberg, 2020; Lyu et al., 2023). Prior work shows that CoT often acts as post-hoc rationalization, generating explanations that diverges from the underlying causal logic (Turpin et al., 2023) or internal attention patterns (Lanham et al., 2023; Barez et al., 2025). Moreover, such behavioral analyses provide limited insight into the specific neurons or circuits driving model predictions.

To overcome the lack of causal rigor in probing (Alain and Bengio, 2018; Belinkov, 2021) and attention analysis (Clark et al., 2019; Vig and Belinkov, 2019), research has pivoted to attribution methods (Sundararajan et al., 2017; Meng et al., 2023) and circuit tracing (Elhage et al., 2021). Recent methods leverages Sparse Autoencoders (SAEs) (Bricken et al., 2023; Cunningham et al., 2023) and Transcoders (Dunefsky et al., 2024) to decompose dense representations into interpretable features (Bloom et al., 2024; Ferrando et al., 2025; Paulo et al., 2025), enabling attributed graphs (Hanna et al., 2025; Marks et al., 2025) to model causal flow. GraphGhost extends this framework to uncover internal structures that affect LLM behavior in both sample and dataset view.

3 Tracing the GraphGhosts

The reasoning behavior of LLMs emerges through successive token generations. Accordingly, existing interpretation methods primarily operate at the token level, attributing individual predictions to specific tokens or activations and providing localized explanations, as illustrated in Figure 2(a) (Ameisen et al., 2025). However, reasoning does not arise from isolated tokens alone, but from structured interactions among multiple tokens

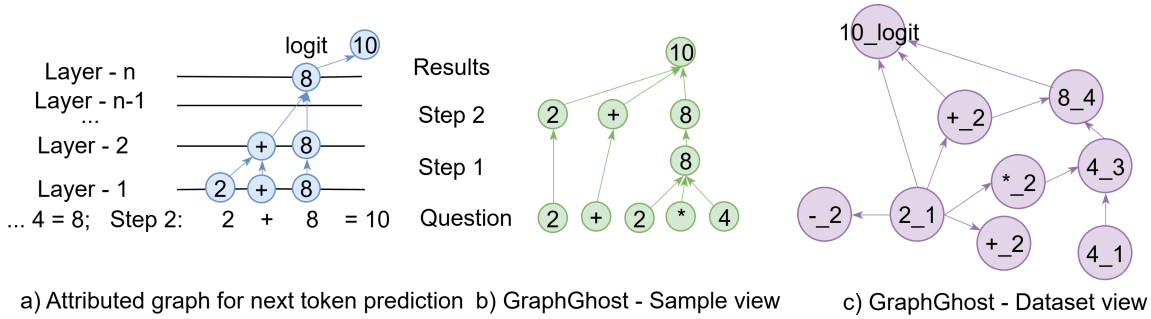


Figure 2: We use a simple example: 2 + 2 * 4. (a) An attributed graph interpreting a single-token prediction. (b) GraphGhost (sample view) aggregates token dependencies along the reasoning chain. (c) GraphGhost (dataset view) captures cross-layer token correlations learned from the corpus. Here, "2_1" denotes token 2 at layer 1; "-_2" does not appear in the sample but is activated from other corpus contexts. Token "2_1" participates in both "2*4" and "2+8" computations.

within a sample and from recurring patterns shared across samples.

To capture these aspects, we introduce two complementary views: a sample view and a dataset view. The sample view models token interactions within a single reasoning instance and formalizes this instance-specific structure as a GraphGhost (Figure 2(b); Section 3.1). The dataset view aggregates such structures across samples to reveal reusable reasoning patterns and is formalized in Section 3.2 (Figure 2(c)).

3.1 Sample View

LLMs have demonstrated strong capabilities in chain-of-thought reasoning. However, whether these reasoning chains faithfully correspond to coherent internal reasoning mechanisms remains unclear. Rather than attempting to answer this question conclusively, we begin with a toy example to illustrate how circuit-tracing methods can be used to probe the internal computations underlying such reasoning behaviors.

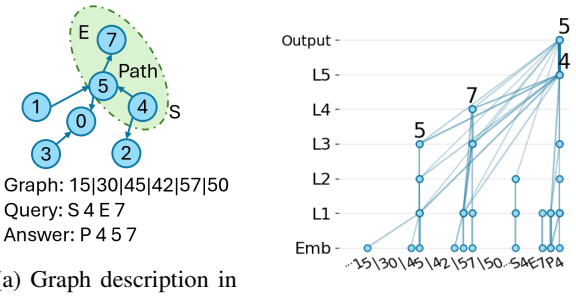
Toy Example on Graph Reasoning Since search paths on a graph naturally decompose reasoning into node-by-node transitions, we adopt the path-reasoning task as a toy example to illustrate how nodes are selected at each step of the reasoning process. Following prior work (Saparov et al., 2025; Wang et al., 2024) that studies LLM reasoning and planning through simplified graph abstractions, we focus on shortest-path search as our canonical setting, as illustrated in Figure 3a.

Graphs are specified as lists of edges, with the symbol "|" used to separate individual edges. In the queries, we denote the start and end nodes using the markers "S" and "E", respectively. In the answers, we use the symbol "P" to indicate that the model is required to generate an explicit reasoning path. we train a 5-layer GPT-2 architecture decoder-only transformer on this task. The trained model achieves 98% accuracy on the test set.To understand how the trained model internally performs path finding, we analyze its inference process using circuit-tracing methods.

Specifically, following (Hanna et al., 2025), we train a transcoder (Dunefsky et al., 2024) for each layer of the transformer. Each transcoder takes the layer-wise hidden activations as input and learns a sparse feature representation that reconstructs these activations, yielding token-level semantic features for each MLP layer. We then replace the original MLP layers with the corresponding transcoders and compute the logit-level contribution weights of each transcoder feature to the final prediction. Using transcoder features as nodes and their logit contribution weights as directed edges, we construct the attributed graph. The resulting attribution graph, visualized in Figure 3b, illustrates how the transformer models the path-finding process. Full experimental details are provided in Appendix B.1.

Circuit tracing reveals that decoder-only transformers revisit the preceding context to identify relevant nodes and predict the next token. From a structural perspective, neighboring nodes are progressively merged in the middle layers. For example, at the third layer, node 4 is merged with node 5, forming a semantic representation of the edge between them. During generation, the final token in the sequence receives the signal that an edge exists between nodes 4 and 5 at layer 3, which leads the model to predict node 5 as the next token. This mechanism differs from human graph reasoning, which explicitly retrieves relations, checks connectivity, and composes shortest paths. We then evaluate this assumption on real LLM reasoning tasks.

GraphGhost in Sample Interpretation In the toy example, we observe that reasoning in basic decoder-only transformers is driven by token-level associations, where previously observed in-context



(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 3: 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.

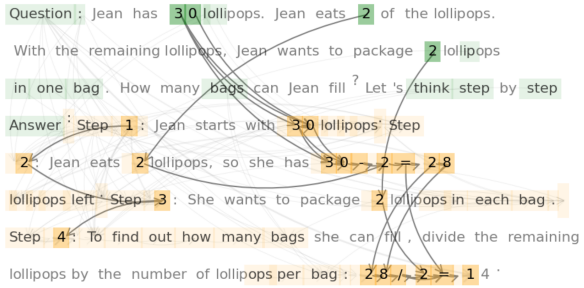


Figure 4: Case study tracing tokens contributing to the answer 14 in Qwen3-0.6B, illustrating how GraphGhost captures the numerical reasoning flow linking number-related tokens to the final prediction.

tokens contribute to information aggregation during token generation, rather than through explicitly modeled logical structures. Motivated by this observation, we merge answer-related circuits to examine how reasoning is internally realized for a specific query in LLMs.

At the sample level, GraphGhost aims to capture the internal structural dependencies that contribute to a model's final prediction. Given a sample s with final answer token t_n, we initialize an empty graph $G^{\text{sample}} = (V, E)$ together with a local edge-weight accumulator W^{local}, which is used to aggregate contribution strengths when the same dependency is discovered multiple times during backward expansion. The graph construction starts from t_n and iteratively expands backward through the model's internal computation.

For each token t in the current frontier that is not part of the question tokens, we apply a circuit tracing procedure to extract the local logit-based attribution graph rooted at t follows the previous paragraph. This step yields a set of contributing tokens as sample-view nodes $\hat{V}$, directed edges $\hat{E}$, and local attribution weights W^{loc}, characterizing how information flows to the prediction of t. Newly discovered tokens are added to the frontier to enable recursive exploration of upstream dependencies.

The nodes and edges from each local attribution graph are by taking the union over nodes and edges, removing duplicates and retaining only the existence of dependency relations. As a result, the sample-level GraphGhost is an unweighted structural graph that captures which token-level dependencies are present, independent of their individual contribution strengths. In parallel, local edge weights are accumulated in W^{local} for each observed edge, which is retained solely for later aggregation at the dataset level. The final output is a sample-specific GraphGhost graph $G^{\text{sample}} = (V, E)$ together with the local edge-weight map W^{local}. The overall algorithm is shown in Algorithm 1 in Appendix A.

The sample view of GraphGhost characterizes step-by-step reasoning in LLMs by modeling structural dependencies among tokens as reasoning unfolds. Here, we show an example in Figure 4. 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: (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.

3.2 Dataset View

The reasoning capabilities of LLMs are learned from large-scale corpora, which motivates the assumption that certain neuron units within LLMs encode reusable token-level connections that support generalization to new reasoning tasks. Accordingly, we introduce the dataset-level view of GraphGhost to explore the implicit structures inside LLMs.

3.2.1 Method

To build the dataset-level of GraphGhost, we aggregate sample-level reasoning structures to reveal recurring internal information-flow patterns shared across multiple samples from datasets. Given a set of samples $\mathcal{S} = \{s_1, \ldots, s_m\}$, we construct a unified graph for the dataset as $G^{\text{data}} = (V^{\text{data}}, E^{\text{data}}, W^{\text{data}})$.

For each sample $s_i \in \mathcal{S}$, we first obtain its sample-level GraphGhost $G^{(i)} = (V^{(i)}, E^{(i)})$ to-

gether with a local edge-weight map W^{local} using the sample-view construction. The dataset-level node and edge sets are formed by merging sample graphs via set union: $V^{\text{data}} = \bigcup_i V^{(i)}$ and $E^{\text{data}} = \bigcup_i E^{(i)}$. In parallel, edge attribution strengths are accumulated across samples. For each edge $e \in E^{(i)}$, we update its dataset-level weight as $W^{\text{data}}(e) \leftarrow W^{\text{data}}(e) + W^{\text{local}}(e)$, where W^{local} captures unnormalized, sample-specific contribution strengths. This accumulation preserves both the frequency and strength of recurring dependencies across samples. After aggregating all samples, we apply row-stochastic normalization to W^{data} to obtain relative edge importance at the dataset level. The resulting graph captures recurring reasoning paths, while the normalized weights quantify their global importance across the dataset. The full procedure is summarized in Algorithm 2 in Appendix A.

The dataset view aims to reveal whether LLMs exhibit recurring internal structures across different tasks and datasets. To this end, we visualize a subgraph extracted from the dataset-level GraphGhost in Figure 5, which aggregates sample-level GraphGhosts constructed from four datasets spanning different domains. In this graph, each node represents a token–layer pair identified by GraphGhost, and node colors indicate the datasets in which the token participates, while mixed colors denote tokens shared across multiple datasets. From this visualization, several consistent patterns emerge. First, certain tokens appear across all four datasets, suggesting shared internal mechanisms. For example, the period symbol "." at layer 22 exhibits strong logit contributions across all datasets, and the discourse marker "So" frequently appears in diverse reasoning chains. Second, domain-specific patterns are also evident: arithmetic-related tokens such as "Adding" and expressions like "6+" are shared between GSM8K and MAWPS, which both involve numerical reasoning. In contrast, some nodes are task-specific; for example, B_{18} appears only in ARC, reflecting choice-based reasoning. This shows that dataset-level GraphGhost captures both shared reasoning primitives and task-specific structures.

3.2.2 Analysis

Beyond visualization, we conduct further graph-based analyses to examine how the structures identified by GraphGhost correspond to LLM behaviors. Specifically, we analyze the structures from two

Figure 5: A subgraph of GraphGhost in Qwen3-0.6B model. We include 4 datasets from different domains.

perspectives: (1) token-merging behavior, quantified by node in-degree; and (2) trigger tokens that initiate reasoning, identified using PageRank. In addition, we present case studies to illustrate how perturbations based on GraphGhost-identified structures affect the progression of the reasoning process.

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). We evaluate these models on three reasoning domains: (i) mathematical reasoning (GSM8K, MAWPS), (ii) logical reasoning (ProntoQA, BoolQA), and (iii) scientific reasoning (ARC-Easy, QASC). We obtain the GraphGhosts for each model and dataset. All experiments are conducted on a single H200 GPU. Each sample will run 0.5-3h, depended on the length of reasoning tokens. Full experimental details are provided in Appendix B.2.

Semantic Merging We begin by analyzing semantic merging behavior using the average in-degree of neuron nodes across layers. We assume that multiple tokens converge within the network, indicating potential semantic consolidation. For instance, in the representation of the number "10", the tokens "1" and "0" exhibit convergent interactions in intermediate layers, consistent with the formation of a unified numerical representation.

In the GraphGhost representation, in-degree quantifies how many contributing nodes are merged into a target node. Accordingly, nodes with higher in-degree correspond to tokens that integrate information from multiple sources and serve as central aggregation points for reasoning-related signals. Based on this, we examine which layers and tokens are preferentially in LLMs by GraphGhost structures.

The statistical results are summarized in Appendix C and visualized in Figure 6. We observe that tokens with high in-degree exhibit clear domain sensitivity. For example, in the ProntoQA dataset, the model is more sensitive to the period

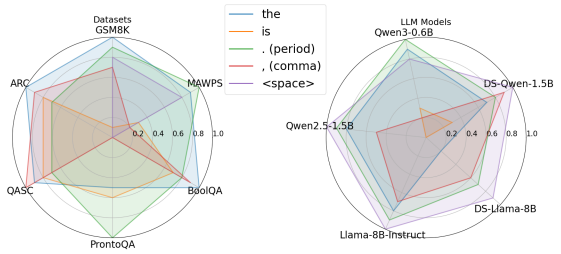


Figure 6: The top merging nodes evaluated by in-degree ratio. Left: The highest score of the in-degree ratio across datasets. Right: The highest score of the in-degree ratio across the models

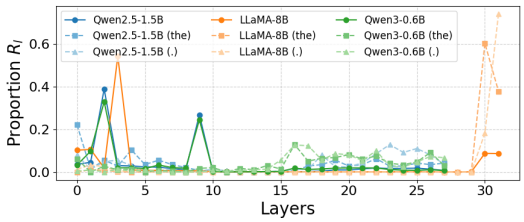


Figure 7: The in-degree proportions on various layers with MAWPS dataset

token than to the comma, suggesting that punctuation plays different semantic roles across reasoning domains. Across models, however, the dominant tokens are largely consistent: special tokens such as <space>, the period, and "the" consistently rank among the top, and this pattern persists even after finetuning.

We further set the *in-degree ratio* to measure the merging behavior across the datasets and models. 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. A higher in-degree ratio R_ℓ implies that layer ℓ concentrates a larger fraction of merging behavior, indicating that many reasoning-related signals converge at this layer. In contrast, a lower ratio suggests that neurons in the layer are less involved in integrating information, playing a more distributive or transitional role in the reasoning process.

As shown in Figure 7, 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.

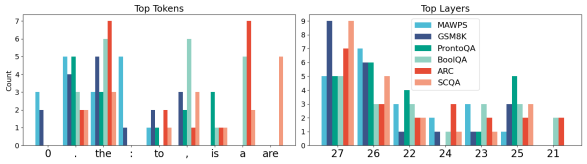


Figure 8: Top contributing tokens and layers in the dataset-level GraphGhost of Qwen3-0.6B across different datasets. Left: The most frequently contributing tokens. Right: The most frequently contributing tokens.

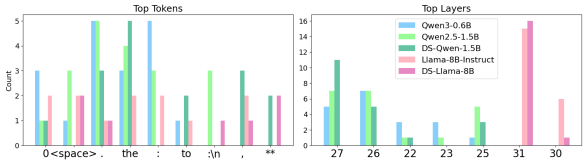


Figure 9: Top contributing tokens and layers across models evaluated by MAWPS dataset. Left: The most frequently contributing tokens. Right: The most frequently contributing tokens.

Reasoning Triggers The reasoning structure is segmented by special tokens such as So, Step, and Wait (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 tokens. 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 8 and Figure 9, and the full details can be found in Appendix C.

Figure 8 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 9 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.

Case Study We further conduct case studies to illustrate how semantic merging and reasoning-trigger effects influence the reasoning process. Based on the analyses in the previous sections,

Case studies

Semantic merging controls 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。...

Reasoning trigger changes logic

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.

Figure 10: Semantic control and reasoning triggers change the model generation processing. In the language-switching case, the generated characters are direct translations of the original answer.

we observe that tokens, such as "the" and ".", are identified as important by both analysis methods, although their functional roles vary across layers. For instance, in Qwen3, these tokens tend to merge in middle layers (e.g., around layer 15), while in higher layers (e.g., beyond layer 25), they often act as reasoning triggers. Accordingly, we design case studies to examine how the roles of such tokens evolve across layers during reasoning.

Specifically, we mute neurons in the LLM when they correspond to a given token at a specified layer. We show that muting neurons in middle layers leads to semantic changes, as illustrated in Figure 10. When the neuron node associated with "the" is deleted at layer 20, the English semantic pathway is disrupted, activating Chinese neurons that share similar semantic signals. As a result, the model produces output in Chinese, effectively translating from the original English text. In the Reasoning trigger 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. The full results are shown in Appendix B.3.

4 Quantitative Experiments

To evaluate whether the structural patterns identified within LLMs are functionally relevant to reasoning behavior, We perform quantitative experiments that examine whether GraphGhost-identified tokens exhibit consistent and measurable significance. We first introduce the evaluation metrics in Section 4.1. The evaluation considers both the sample-level view (Section 4.2) and the dataset-level view (Section 4.3).

4.1 Metrics

We conduct quantitative evaluations using Fidelity, InFidelity, and Sparsity metrics to assess whether model predictions remain consistent with the original, non-perturbed outputs (Yeh et al., 2019). In practice, we use gemini-2.5-flash-lite to determine whether two generated predictions are semantically identical. Sparsity measures the proportion of features identified as important (Guidotti et al., 2018). In the sample view, we evaluate contribution tokens using both Sparsity and Fidelity. Formally, $\text{Sparsity} = \frac{|C|}{|T|}$, $\text{Fidelity}_T = \mathbb{1}(LLM(C),\, LLM(T))$, where C denotes the set of contributing tokens, T denotes the full token set, and $\mathbb{1}(\cdot,\cdot)$ is an indicator function that returns 1 if the two outputs are identical and 0 otherwise. In the dataset view, we evaluate whether muting the identified nodes affects the model's prediction and report InFidelity, defined as $\text{InFidelity} = 1 - \text{Fidelity} = \mathbb{1}\left(LLM(T),\, \hat{LLM}(T)\right)$, which captures cases where muting the selected nodes in $\hat{LLM}(\cdot)$ results in a change in the model's output.

4.2 Contribution Token Selection in GraphGhost: Sample View

To evaluate whether GraphGhost can identify tokens that are functionally important for the reasoning process, we compare it against attention-based explanation methods (Clark et al., 2019; Liu et al., 2024) and random baselines. Instead of analyzing individual tokens, we adapt the same pipeline as GraphGhost to evaluate its effectiveness. The detailed settings are provided in Appendix B.4.

The comparative results are summarized in Figure 11. For a detailed breakdown of the performance for each individual model without averaging, we provide the complete results in Appendix D (Figure 15). GraphGhost consistently outperforms both random and attention-based baselines. This advantage is most pronounced on mathematical reasoning tasks (MAWPS and GSM8K), where

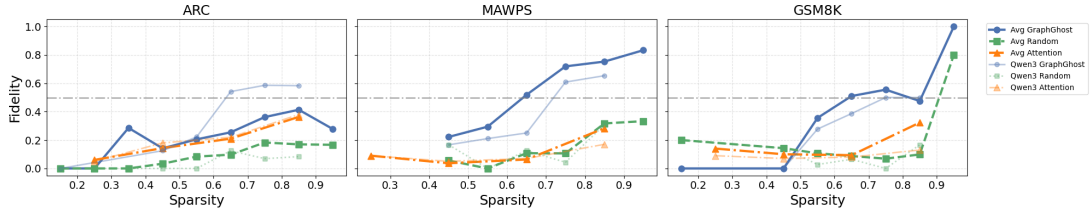


Figure 11: Sparsity Vs. Fidelity in various datasets and models.

the attention baseline performs comparably to random noise, indicating its failure to capture essential reasoning structures. In contrast, GraphGhost maintains high fidelity even under low sparsity constraints. Moreover, on the challenging ARC dataset, GraphGhost demonstrates superior robustness; notably, it is the only method that surpasses the 0.5 fidelity threshold on Qwen3. These results suggest that while attention primarily captures surface-level correlations, GraphGhost is able to trace deeper, multi-hop structural dependencies that are critical for complex reasoning.

4.3 Behaviors Controlled by Critical Nodes in GraphGhost: Dataset View

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.

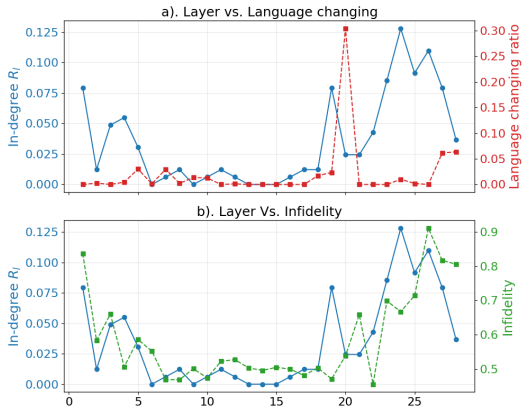


Figure 12: The impact of tokens on various layers. We mute the period token '.' for each layer.

	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

We first 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 in middle layers. Furthermore, as shown in Figure 12(b), Infidelity increases in the upper layers. This indicates that neuron nodes in higher layers are more likely to connect directly to logit neurons, thereby altering the underlying reasoning logic. This observation is consistent with the findings in Section 3.2.2, where perturbations in middle layers primarily affect semantic expression, whereas perturbations in higher layers tend to modify the reasoning logic itself.

We also evaluate the important tokens ranked by PageRank algorithm. 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 much higher influence, indicating that PageRank effectively identifies important nodes in GraphGhost. Larger models are more stable than smaller ones; Llama-8B achieves an InFidelity score of 0.28, whereas Qwen3 reaches 0.58. Notably, the <space> token is the most influential, especially in higher layers.

5 Conclusion

We present GraphGhost, a graph-based framework that reveals the internal structures underlying LLM reasoning. By linking token interactions and neuron activations to graph properties, GraphGhost shows how semantic merging and reasoning triggers shape model behavior. Our results demonstrate that LLM reasoning is governed by structured internal graphs, offering a principled path toward mechanistic interpretability.

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.

Potential Risk

A potential risk of our method is that the extracted graph structures may be over-interpreted as definitive explanations of model behavior, while they only provide approximate views of internal information flow. We caution against such misuse and position our approach as a complementary analytical tool.

References

Guillaume Alain and Yoshua Bengio. 2018. Understanding intermediate layers using linear classifier probes. *Preprint*, arXiv:1610.01644.

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*.

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.

Yonatan Belinkov. 2021. Probing classifiers: Promises, shortcomings, and advances. *Preprint*, arXiv:2102.12452.

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*.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, and 95 others. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, and 1 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2.

Xi Chen, Aske Plaat, and Niki van Stein. 2025. How does chain of thought think? mechanistic interpretability of chain-of-thought reasoning with sparse autoencoding. *Preprint*, arXiv:2507.22928.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention. *Preprint*, arXiv:1906.04341.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

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

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. *Preprint*, arXiv:2305.18654.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Javier Ferrando, Oscar Obeso, Senthooran Rajamanoharan, and Neel Nanda. 2025. Do i know this entity? knowledge awareness and hallucinations in language models. *Preprint*, arXiv:2411.14257.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. 2025. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Preprint*, arXiv:2301.04709.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box

models. *ACM computing surveys (CSUR)*, 51(5):1–42.

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.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *Preprint*, arXiv:2004.03685.

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.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *Preprint*, arXiv:2307.13702.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *Preprint*, arXiv:2206.14858.

Fuxiao Liu, Paiheng Xu, Zongxia Li, Yue Feng, and Hyemi Song. 2024. Towards understanding in-context learning with contrastive demonstrations and saliency maps. *Preprint*, arXiv:2307.05052.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *Preprint*, arXiv:2301.13379.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2025. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *Preprint*, arXiv:2403.19647.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023. Locating and editing factual associations in gpt. *Preprint*, arXiv:2202.05262.

Navapat Nananukul, Yue Zhang, Ryan Lee, Eric Boxer, Jonathan May, Vibhav Giridhar Gogate, Jay Pujara, and Mayank Kejriwal. 2025. Logicalthought: Logic-based ontological grounding of llms for high-assurance reasoning. *Preprint*, arXiv:2510.01530.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. *Preprint*, arXiv:2301.05217.

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

Tushar Pandey, Ara Ghukasyan, Oktay Goktas, and Santosh Kumar Radha. 2025. Adaptive graph of thoughts: Test-time adaptive reasoning unifying chain, tree, and graph structures. *Preprint*, arXiv:2502.05078.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. 2025. Automatically interpreting millions of features in large language models. *Preprint*, arXiv:2410.13928.

Abulhair Saparov, Srushti Pawar, Shreyas Pimpalgaonkar, Nitish Joshi, Richard Yuanzhe Pang, Vishakh Padmakumar, Seyed Mehran Kazemi, Najoung Kim, and He He. 2025. Transformers struggle to learn to search. *Preprint*, arXiv:2412.04703.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *Preprint*, arXiv:1703.01365.

Ala N. Tak, Amin Banayeeanzade, Anahita Bolourani, Mina Kian, Robin Jia, and Jonathan Gratch. 2025. Mechanistic interpretability of emotion inference in large language models. *Preprint*, arXiv:2502.05489.

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.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. *Preprint*, arXiv:1906.04284.

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. *Preprint*, arXiv:2405.09220.

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.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. 2019. On the (in) fidelity and sensitivity of explanations. *Advances in neural information processing systems*, 32.

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 Appendix: Algorithm

Here we present the algorithm of construct the sample and dataset view of GraphGhost in Algorithm 1 and Algorithm 2.

Algorithm 1 GraphGhost Construction (Sample View)

Input: A sample s containing a final answer token t_n
Output:
- Sample-level GraphGhost $G^{\text{sample}} = (V, E)$ (unweighted)
- A local edge-weight map W^{local} (for dataset aggregation only)

Initialize $V = \emptyset$, $E = \emptyset$, and $W^{\text{local}}(e) = 0$ for all edges e
Initialize frontier $T = \{t_n\}$
for each token $t \in T$ **do**
  if t is not part of the question tokens **then**
    # Step 1: Local logit-based attribution graph
    $(\hat{T}, G'_A) \leftarrow \textsc{CircuitTracer}(t)$ ▷ $G'_A = (\hat{V}, \hat{E}, W^{\text{loc}})$
    $T \leftarrow T \cup \hat{T}$
    # Step 2: Add edges to sample-level graph (structure only)
    $V \leftarrow V \cup \hat{V}$
    $E \leftarrow E \cup \hat{E}$ ▷ unweighted sample graph
    # Step 3: Accumulate local weights for dataset view
    for each edge $e \in \hat{E}$ **do**
      $W^{\text{local}}(e) \leftarrow W^{\text{local}}(e) + W^{\text{loc}}(e)$
    end for
  end if
end for
return $(G^{\text{sample}} = (V, E),\ W^{\text{local}})$

Algorithm 2 GraphGhost Construction (Dataset View)

Input: A set of samples $\mathcal{S} = \{s_1, \ldots, s_m\}$
Output: Dataset-level GraphGhost $G^{\text{data}} = (V^{\text{data}}, E^{\text{data}}, W^{\text{data}})$
Initialize $V^{\text{data}} = \emptyset$, $E^{\text{data}} = \emptyset$, and $W^{\text{data}}(e) = 0$
for each sample $s_i \in \mathcal{S}$ **do**
  $(G^{(i)}, W^{\text{local}}) \leftarrow \textsc{GraphGhost-SampleView}(s_i)$
  $(V^{(i)}, E^{(i)}) \leftarrow G^{(i)}$
  # Merge sample graph structure
  $V^{\text{data}} \leftarrow V^{\text{data}} \cup V^{(i)}$
  $E^{\text{data}} \leftarrow E^{\text{data}} \cup E^{(i)}$
  # Accumulate edge weights across samples
  for each edge $e \in E^{(i)}$ **do**
    $W^{\text{data}}(e) \leftarrow W^{\text{data}}(e) + W^{\text{local}}(e)$
  end for
end for
return $G^{\text{data}} = (V^{\text{data}}, E^{\text{data}}, W^{\text{data}})$

B Appendix: Experiment setting

B.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 where the validation loss is the minimal.

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.

B.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.

B.3 Case studies

We show the full case studies of how neuron nodes change the LLMs behaviors in Figure 14 and Fig-

		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>"

ure 13. We show that muting neurons in middle layers leads to semantic changes, as illustrated in Figure 13. When the neuron node associated with "the" is deleted at layer 20, the English semantic pathway is disrupted, activating Chinese neurons that share similar semantic signals. As a result, the model produces output in Chinese, effectively translating from the original English text. Furthermore, in the word-expression change case, deleting the neuron associated with "." at layer 17 causes the token "/" to replace "÷".

In Figure 14, we demonstrate how GraphGhost influences reasoning by altering key steps or driving the generation into loops. 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. 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.

B.4 Attention-based Baseline Implementation

To adapt the standard attention analysis (Clark et al., 2019) for reasoning tasks, we implement a specific Answer-to-Input saliency measure. We first calculate the mean attention matrix $\bar{\mathbf{A}}$ by averaging weights across all heads in the final layer. The importance score S_i for each input token i is then derived by accumulating the attention it receives from the entire generated answer sequence:

$$S_i = \sum_{j \in \text{Answer}} \bar{\mathbf{A}}_{j,i}$$

This score represents the total attention mass focused on input i during the reasoning generation. Input tokens are ranked by S_i, and the top-$k\%$ are retained to construct the perturbed prompt for fidelity evaluation, following the same protocol as GraphGhost.

C 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 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

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

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 13: Semantic control in middle layers modulates changes in semantic representations. In the language-switching case, the generated characters are direct translations of the original answer.

Change the reasoning structure

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 ...

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.

Figure 14: Reasoning triggers changes on the reasoning structures

D Additional Quantitative Results

Figure 15 presents the detailed fidelity-sparsity curves for all evaluated models (DS-Qwen-1.5B, Qwen3-0.6B, and Qwen2.5-1.5B) across the three datasets. Unlike the aggregated results in the main text (Figure 11), this figure illustrates the individual performance trajectories for each model without averaging.

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.

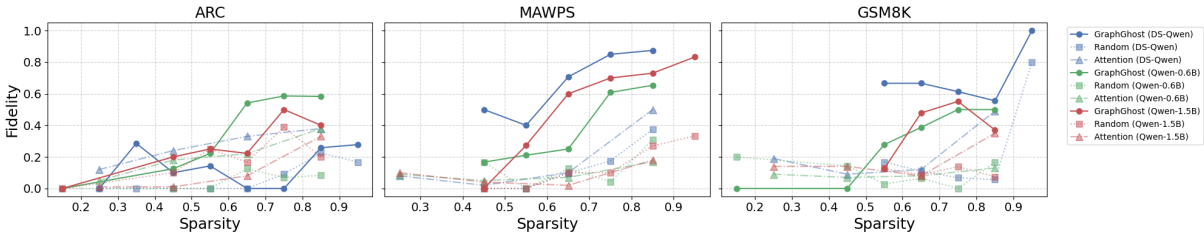


Figure 15: Detailed fidelity-sparsity curves for individual models across ARC, MAWPS, and GSM8K datasets. This figure shows the non-averaged performance for GraphGhost compared to Random and Attention baselines.