Dual Path Attribution: Efficient Attribution for SwiGLU-Transformers through Layer-Wise Target Propagation

Lasse M. Jantsch [*1] Dong-Jae Koh [*1] Seonghyeon Lee [1] Young-Kyoon Suh [1]

Abstract

Understanding the internal mechanisms of transformer-based large language models (LLMs) is crucial for their reliable deployment and effective operation. While recent efforts have yielded a plethora of attribution methods attempting to balance faithfulness and computational efficiency, dense component attribution remains prohibitively expensive. In this work, we introduce *Dual Path Attribution* (DPA), a novel framework that faithfully traces information flow on the frozen transformer in one forward and one backward pass without requiring counterfactual examples. DPA analytically decomposes and linearizes the computational structure of the SwiGLU Transformers into distinct pathways along which it propagates a targeted unembedding vector to receive the effective representation at each residual position. This target-centric propagation achieves $O(1)$ time complexity with respect to the number of model components, scaling to long input sequences and dense component attribution. Extensive experiments on standard interpretability benchmarks demonstrate that DPA achieves state-of-the-art faithfulness and unprecedented efficiency compared to existing baselines.

1. Introduction

Transformer-based large language models (LLMs) are the dominant model paradigm for general-purpose AI; however, the internal mechanisms driving this performance remain poorly understood. A central challenge is attribution, identifying which inputs or internal components are responsible for a model's predictions.

While a plethora of attribution approaches were applied to LLMs (Ferrando et al., 2024), they are often computationally expensive (Sundararajan et al., 2017; Conmy et al., 2023), ignore important model components (Hong et al., 2025), or suffer from approximation errors or noise (Shrikumar et al., 2019; Balduzzi et al., 2018). One promising direction is decomposition based approaches attempting to fully linearize the computational graph of the transformers.

We propose *Dual Path Attribution* (DPA), a decomposition-based framework for efficient input and component attribution. By propagating the unembedding vector of the target token through the frozen transformer, DPA approximates an effective target for each residual position in one forward and one backward pass. While DPA can be applied to all transformer architectures, this paper focuses on the SwiGLU transformer, as it allows for the decomposition of its bilinear interactions into distinct pathways: a *content* pathway that propagates representations through the value and up projections, and a *control* pathway that governs their routing and transformation through attention and gate scores. Experiments on standard interpretability benchmarks demonstrate the faithfulness of DPA for both input and component attribution. Figure 1 shows a high-level overview of our approach. Our contributions are summarized as follows:

- We introduce *Dual Path Attribution* (DPA), a novel decomposition based framework, efficiently attributing logit scores in one forward and one backward pass without the need for counterfactual examples.
- By leveraging the bilinear structure of the SwiGLU Transformers, we show that the computational graph is decomposable in distinct control and content pathways.
- Through extensive experiments on standard interpretability benchmarks, we demonstrate that DPA achieves superior faithfulness and efficiency compared to state-of-the-art baselines, effectively identifying causal mechanisms.

All relevant artifacts are publicly available on GitHub[1].

[*]Equal contribution [1]Department of Computer Science and Engineering, Kyungpook National University, Daegu, Korea. Correspondence to: Seonghyeon Lee <sh0416@knu.ac.kr>, Young-Kyoon Suh <yksuh@knu.ac.kr>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

[1]https://github.com/lab-paper-code/dual_path_attribution

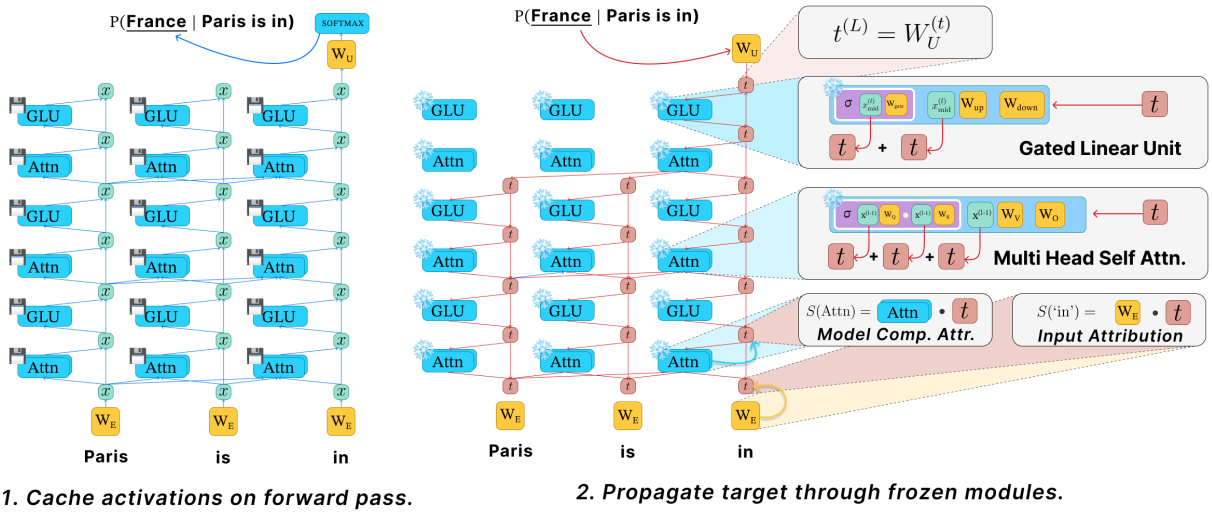

Figure 1. Overview of the *Dual Path Attribution* (DPA) framework for efficient input and component attribution. *DPA* operates in two stages: (1) *One-pass forward execution*, in which the model processes the input once while caching activations required for the back propagation; and (2) *top-down target propagation*, where the unembedding vector of the targeted token is recursively propagated backward through Transformer modules to identify an effective target at each residual position.

2. Related Work

Attribution research for Transformer-based language models broadly aims to localize the sources of model behavior, either by attributing predictions to input tokens or by identifying responsible internal components (Ferrando et al., 2024).

Input attribution. Gradient-based input attribution approaches use the first order Taylor-Expansion as importance signal approximating prediction sensitivity (Simonyan et al., 2013; Denil et al., 2014; Li et al., 2015). *Input x Gradient* (Denil et al., 2015) calculates the dot product between the embedding vector and the gradient, including input magnitude. While gradient based approaches suffer from gradient saturation and shattering (Shrikumar et al., 2019; Balduzzi et al., 2018), approaches like *Integrated Gradients* (Sundararajan et al., 2017) or *SmoothGrad* (Smilkov et al., 2017) were shown effective to denoising the gradients. *Layer-wise Relevance Propagation* (LRP) (Bach et al., 2015) conserves the importance signal by applying local propagation rules (Voita et al., 2021; Ali et al., 2022; Achtibat et al., 2024).

Another line of previous work analyzes the flow of contextual information. Early approaches interpreted raw attention weights, which were later shown to be unreliable indicators of importance (Jain & Wallace, 2019). Subsequent methods incorporated value-vector norms (Kobayashi et al., 2020; 2021) or propagated attention across layers using rollout techniques (Abnar & Zuidema, 2020b; Ferrando et al., 2022; Modarressi et al., 2022). More recent work favors decomposition-based approaches that explicitly expand the model's linear structure, yielding more faithful attribution of attention-mediated effects (Modarressi et al., 2023; Kobayashi et al., 2024; Hong et al., 2025).

Component-level attribution. Beyond input tokens, recent studies have focused on attributing internal components to the model output. *Direct Logit Attribution* (DLA) (Elhage et al., 2021; nostalgebraist, 2020) exploits the additive structure through the residual connection to measure a component's direct contribution to output logits, and has been applied to both attention heads (Ferrando et al., 2023) and individual neurons (Geva et al., 2022). DLA, however, does not capture indirect effects of downstream layers, leading to decreased interpretability in early layers.

Causal intervention methods (i.e., activation patching) resolve this limitation by measuring direct and indirect effects via counterfactual forward passes (Vig et al., 2020; Meng et al., 2022; Wang et al., 2022; Zhang & Nanda, 2023). While highly faithful, activation patching is computationally expensive for dense attribution as it requires separate forward passes for each component or interaction.

Recently, automated circuit discovery – the detection of component subsets responsible for model behavior – has seen causal intervention (Conmy et al., 2023), gradient (Nanda, 2023; Syed et al., 2023; 2024), LRP (Jafari et al., 2025), and contextual mixing (Ferrando & Voita, 2024) based approaches.

3. Transformer Architecture

This section reviews the Transformer components relevant to our method and establishes the notation used throughout the paper. We focus on the Swish-Gated Linear Unit (SwiGLU) (Shazeer, 2020) architecture, as it is the current state-of-the art for dense Transformers. A general overview of decoder-only Transformers is provided in Appendix A. We formalize our exposition based on the Llama2-style architecture (Touvron et al., 2023; Grattafiori et al., 2024), though our approach directly generalizes to other SwiGLU-based Transformers with minimal adjustment.

3.1. Multi-Head Self-Attention

The multi-head self-attention (MHSA) module differs from the original Transformer (Radford et al., 2019) by employing Root Mean Square Normalization (RMSNorm) (Zhang & Sennrich, 2019) and Rotary Positional Embeddings (RoPE) (Su et al., 2021). Given an input sequence representation $\mathbf{X} \in \mathbb{R}^{T\times d}$, MHSA computes attention over queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$ across H independent heads.

The output of MHSA is formed by concatenating head-wise outputs and projecting them back to the residual stream via an output matrix $\mathbf{W}_O \in \mathbb{R}^{d\times d}$:

$$\mathrm{MHSA}(\mathbf{X}) = \alpha\big(\tilde{\mathbf{X}}\mathbf{W}_V\big)\mathbf{W}_O,$$

where $\tilde{\mathbf{X}}$ denotes the RMS-normalized input, $\mathbf{W}_V \in \mathbb{R}^{d\times H\times d_h}$ is the value projection, and d_h is the head dimension. The attention weights α are computed as:

$$\alpha = \mathrm{softmax}\left(\frac{R_Q(\tilde{\mathbf{X}}\mathbf{W}_Q)\, R_K(\tilde{\mathbf{X}}\mathbf{W}_K)^\top}{\sqrt{d_h}}\right),$$

where $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d\times H\times d_h}$ are the query and key projections, and $R(\cdot)$ denotes the application of RoPE.

3.2. Gated Linear Units

The feed-forward network in LLaMA-style models adopts the SwiGLU formulation, which separates computation into a gating and a value pathway. Given an input $\mathbf{X}$, the GLU transformation is defined as:

$$\mathrm{GLU}(\mathbf{X}) = \Big(\sigma(\tilde{\mathbf{X}}\mathbf{W}_G) \odot (\tilde{\mathbf{X}}\mathbf{W}_U)\Big)\mathbf{W}_D,$$

where $\sigma(\cdot)$ denotes the SiLU (Swish) activation, $\odot$ is element-wise multiplication, and $\mathbf{W}_G, \mathbf{W}_U \in \mathbb{R}^{d\times d_{\text{ffn}}}$ and $\mathbf{W}_D \in \mathbb{R}^{d_{\text{ffn}}\times d}$ are the gate, up, and down projection matrices, respectively.

3.3. Component-wise Decomposition of Transformer Modules

A key property exploited by DPA is the linear separability of the transformer blocks (Modarressi et al., 2023; Kobayashi et al., 2021), allowing for decomposition into independent model component contributions; attention heads and neurons for MHSA and GLU respectively.

For MHSA, the output can be expressed as the sum of head-wise contributions:

$$\mathrm{MHSA}(\mathbf{X}) = \sum_{h=1}^{H} \alpha^{(h)}\big(\tilde{\mathbf{X}}\mathbf{W}_V^{(h)}\big)\mathbf{W}_O^{(h)},$$

where $\mathbf{W}_V^{(h)}$ and $\mathbf{W}_O^{(h)}$ denote the value and output projections of head h.

Similarly, the GLU output can be decomposed into contributions from individual neurons:

$$\mathrm{GLU}(\mathbf{X}) = \sum_{i=1}^{d_{\text{ffn}}} \Big(\sigma(\tilde{\mathbf{X}}\mathbf{w}_G^{(i)}) \odot (\tilde{\mathbf{X}}\mathbf{w}_U^{(i)})\Big)\mathbf{w}_D^{(i)},$$

where $\mathbf{w}_G^{(i)}$ and $\mathbf{w}_U^{(i)}$ denote the i-th columns of $\mathbf{W}_G$ and $\mathbf{W}_U$, and $\mathbf{w}_D^{(i)}$ denotes the i-th row of $\mathbf{W}_D$. Each term corresponds to the contribution of a single GLU neuron to the residual stream.

4. Problem Statement

Our objective is to *quantify how much a given model component* $\mathbf{x}$ *contributes to a target model output* $\mathbf{t}$. Formally, we seek to assign an attribution score $S_t(\mathbf{x})$ that measures the relevance of $\mathbf{x}$ to a target token prediction $\mathbf{t}$. Let $\mathbf{x}_i^{(0)} \in \mathbb{R}^d$ denote the embedding vector at token position i, and let $\mathbf{t}_j \in \mathbb{R}^d$ be the j-th row of the unembedding matrix W_{UE}, corresponding to the direction in latent space that increases the logit $logit_i^{(j)}$ of token $t^{(j)} \in \mathcal{V}$ at position i. Owing to the additive structure of the residual stream, $\mathbf{x}_i^{(0)}$ contributes directly to the output logit, yielding the direct attribution score:

$$S_{t,\text{direct}}(\mathbf{x}_i^{(0)}) = (\mathbf{x}_i^{(0)})^\top \mathbf{t}_j.$$

However, this direct contribution does not exist for embeddings at position $i' < i$ and further—particularly for early-layer components such as embeddings—accounts for only a small fraction of their true influence, as most effects are mediated through downstream transformations (Belrose et al., 2023). The full attribution can be formalized as:

$$S_t(\mathbf{X}^{(0)}) = f(\mathbf{x}_i^{(0)})^\top \mathbf{t}_j,$$

where $f(\cdot)$ denotes the downstream processing of $\mathbf{x}_i^{(0)}$ with an otherwise frozen network. While faithful (Hong et al.,

2025), this approach requires a separate forward pass per component to evaluate $f(\mathbf{x}_i^{(0)})$, making dense attribution prohibitively expensive.

To overcome this limitation, we invert the attribution perspective: rather than passing each component forwards, we propagate the target backwards. Specifically, we reformulate the attribution score as:

$$S_t(\mathbf{x}_i^{(0)}) = (\mathbf{x}_i^{(0)})^\top f^{-1}(\mathbf{t}_j) = (\mathbf{x}_i^{(0)})^\top \mathbf{t}_j^{(0)},$$

where $\mathbf{t}_j^{(0)}$ denotes the *effective target* at layer 0. This effective target is computed recursively via:

$$\mathbf{t}_{\text{mid}}^{(l)} = \mathbf{t}^{(l+1)} + g_{\text{GLU}}^{(l)}\big(\mathbf{t}^{(l+1)}\big),$$
$$\mathbf{t}_j^{(l)} = \mathbf{t}_{\text{mid}}^{(l)} + g_{\text{MHSA}}^{(l)}\big(\mathbf{t}_{\text{mid}}^{(l)}\big),$$

where $g_{\text{GLU}}^{(l)}(\cdot)$ and $g_{\text{MHSA}}^{(l)}(\cdot)$ denote linearized inverse transformations of the $\text{GLU}^{(l)}$ and $\text{MHSA}^{(l)}$ blocks, respectively.

Under this formulation, attribution requires only a single forward pass to cache activations and a single backward pass to propagate the effective target. In the following we describe the inversion and linearization process. In Appendix B, we present theoretical and empirical complexity analyses of the proposed method, comparing it with several baseline methods.

5. Dual Path Attribution

Dual Path Attribution (DPA) linearizes and inverts MHSA and GLU by separately inverting the different pathways through the otherwise frozen module. To prevent the explosion of the target norm and sustain completeness[2], each inversion is further regularized by a weighting parameter μ. Formal derivations of these transformations are provided in Appendix D.

5.1. Neuron Target Propagation

Consider a GLU neuron n with activation $\alpha^{(n)}$. To propagate the effective target backward through the GLU, we approximate the inverse transformation $\mathbf{g}_{\text{GLU}}^{(n)}(\cdot)$ as a weighted combination of the up- and gate-projection paths, assuming frozen activations:

$$\mathbf{g}_{\text{GLU}}^{(n)}(\mathbf{t}) = \mu_{\text{up}} \cdot \mathbf{g}_{\text{up}}^{(n)}(\mathbf{t}) + \mu_{\text{gate}} \cdot \mathbf{g}_{\text{gate}}^{(n)}(\mathbf{t}),$$

where $\mu_{\text{up}} + \mu_{\text{gate}} = 1$.

[2]While DPA generally supports completeness, our choice of softmax linearization breaks it for the sake of a robust gradient.

Up-projection. With frozen activations, the up-projection-pathway is linear and corresponds to the composition of the up- and down-projection weights, $\mathbf{W}_{UD}^{(n)} = \mathbf{w}_U^{(n)}\mathbf{w}_D^{(n)}$. The propagated target is given by:

$$\mathbf{g}_{\text{up}}^{(n)}(\mathbf{t}) = \underbrace{\frac{\mathbf{w}_U^{(n)} \odot \boldsymbol{\gamma}}{\sigma}}_{\text{effective weight } \tilde{\mathbf{w}}_U^{(n)}} \cdot \alpha^{(n)} \cdot \underbrace{\big((\mathbf{w}_D^{(n)})^\top \mathbf{t}\big)}_{\text{gradient scalar } \lambda_{\text{GLU}}},$$

where $\boldsymbol{\gamma}$ and σ denote the RMSNorm parameters. For clarity, we refer to such normalized weights as *effective weights* $\tilde{\mathbf{w}}$.

Gate-projection. Due to the non-linearity, the gate-projection-pathway requires linearization to be inverted. We approximate the local gradient using the ratio $\alpha^{(n)}/s^{(n)}$, where $s^{(n)}$ denotes the pre-activation value:

$$\mathbf{g}_{\text{gate}}^{(n)}(\mathbf{t}) = \tilde{\mathbf{w}}_G^{(n)} \cdot \frac{\alpha^{(n)}}{s^{(n)}} \cdot \lambda_{\text{GLU}}.$$

5.2. Attention Head Target Propagation

For an attention head h, target propagation is more involved due to token mixing softmax activation. We decompose the inverse MHSA transformation into contributions from the query, key, and value paths, and combine them via weighted aggregation:

$$\mathbf{g}_{\text{head}}^{(h)}(\mathbf{t}) = \mu_Q \cdot \mathbf{g}_Q^{(h)}(\mathbf{t}) + \mu_K \cdot \mathbf{g}_K^{(h)}(\mathbf{t}) + \mu_V \cdot \mathbf{g}_V^{(h)}(\mathbf{t}),$$

where $\mu_Q + \mu_K + \mu_V = 1$.

Value path. The value path is linear, given fixed attention weights, and is defined as:

$$\mathbf{g}_V^{(h)}(\mathbf{t}) = \tilde{\mathbf{W}}_V^{(h)} \cdot \underbrace{(\mathbf{W}_O^{(h)})^\top \mathbf{t}}_{\text{grad. vec. } \boldsymbol{\lambda}_{\text{MHSA}}} \cdot \boldsymbol{\alpha}^{(h)}.$$

Query and key paths. Propagating targets through the query and key paths requires tracing gradients through the softmax operation. Unlike GLU activations, softmax outputs depend jointly on all attention scores and remain non-zero even for zero inputs, making naive linearization unstable. Following Achtibat et al. (2024), we adopt a Taylor-based decomposition of the softmax, which provides a stable—though non-conservative—approximation:

$$\mathbf{g}_{Q,i}^{(h)}(\mathbf{t}) = \tilde{\mathbf{W}}_Q^{(h)} \sum_{j \le i} \delta_{ij} \cdot R_{Q,i}^{-1} \left(\frac{R_{K,j}(\tilde{\mathbf{x}}_j \mathbf{W}_K^{(h)})}{\sqrt{d_h}} \right)^\top,$$

$$\mathbf{g}_{K,j}^{(h)}(\mathbf{t}) = \tilde{\mathbf{W}}_K^{(h)} \sum_{i \ge j} \delta_{ij} \cdot R_{K,j}^{-1} \left(\frac{R_{Q,i}(\tilde{\mathbf{x}}_i \mathbf{W}_Q^{(h)})}{\sqrt{d_h}} \right)^\top,$$

where the scalar interaction term δ_{ij} captures the sensitivity of the softmax output:

$$\delta_{ij} = \alpha_{ij}\left(\tilde{\mathbf{x}}_j \mathbf{W}_V^{(h)} - \boldsymbol{\mu}_i^{(h)}\right) \bullet \boldsymbol{\lambda}_{\text{MHSA}}.$$

Here, $\boldsymbol{\mu}_i^{(h)}$ denotes the expected intermediate activation vector of head h at token position i.

6. Experiments

We now evaluate the proposed DPA approach through experiments assessing attribution *faithfulness* and *sensitivity* to the weighting parameters μ.

6.1. Attribution Faithfulness

RQ 1: *Does DPA attribute input features (tokens) and internal component importance faithfully?*

The attribution faithfulness experiments quantify how well DPA identifies components that are causally relevant to a model's prediction. Following prior work (DeYoung et al., 2020), we evaluate faithfulness using two complementary metrics: *comprehensiveness* (disruption) and *sufficiency* (recovery). Both metrics are computed by ablating or retaining the top-K important components and measuring the relative change in the probability of the correct next-token:

$$1 - \Delta p^{(K)} = 1 - \frac{p(y \mid x) - p(y \mid x^{(K)})}{p(y \mid x)},$$

where $x^{(K)}$ denotes the input or internal state after ablating the top-K components.

Input Attribution. For token-level attribution, we compute effective targets $\mathbf{t}_i^{(0)}$ by propagating the unembedding vector of the correct answer token backward to the input layer. Token importance is then obtained via the dot product between $\mathbf{t}_i^{(0)}$ and the corresponding embedding vectors.

Datasets and Models. We evaluate our approach on Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Llama-2-7B-Chat (Touvron et al., 2023), Qwen3-4B-Instruct (Yang et al., 2025), Mistral-7B-Instruct (Jiang et al., 2023), and Qwen2.5-32B-Instruct (Yang et al., 2024). Experiments are conducted on three representative datasets spanning different attribution scenarios: (1) *Known 1000* (Meng et al., 2022) for factual sequence completion, (2) *SQuAD v2.0* (Rajpurkar et al., 2018) for context-supported question answering, and (3) *IMDb* (Maas et al., 2011) for sentiment analysis. See Appendix E for additional information on the datasets and preprocessing.

Baselines. We compare DPA against a broad range of attribution baselines, including gradient-based methods (*Gradient* at input, *Input×Gradient* (Denil et al., 2014), and *Integrated Gradients* (Sundararajan et al., 2017)), attention-based methods (*Last-Layer Attention* and *Mean Attention* on the last token, as well as *Attention Rollout* (Abnar & Zuidema, 2020a)), context-mixing method *Information Flow Routs* (Ferrando & Voita, 2024), and the decomposition-based method *DePass* (Hong et al., 2025).

Evaluation Setup. For the Known 1000 dataset, we assess faithfulness by ablating individual input tokens, excluding special tokens. For SQuAD v2.0 and IMDb dataset, we employ a context- or review-level ablation strategy, in which tokens within the supporting paragraph or review are ablated while the remaining prompt remains intact. This setup mitigates artifacts caused by truncating incomplete sentences immediately preceding the target token, yielding a cleaner faithfulness signal. Additional setup information for reproducibility are located in Appendix F.

Results. As shown in Table 1 and Figure 2, DPA consistently outperforms all baselines on total faithfulness. While DPA is outperformed by simple attention based metrics for Known 1000 (Last Layer Attention - 5.26) and SQuAD v2.0 (Mean Attention - 17.06), attribution to the more complex notion of sentiment shows the limitation of those approaches. Figure 4 shows a qualitative comparison of a subset of the baseline methods on the IMDB dataset. Results for Llama2, Qwen3, Qwen2.5, and Mistral are located in Appendix G.

Module Component Attribution. We further assess attribution faithfulness at the level of internal model components, focusing on individual attention heads and MLP neurons.

Datasets and Models. Experiments are conducted on the Indirect Object Identification (IOI) task (Wang et al., 2022), a widely used benchmark for circuit discovery, as well as the Known 1000 dataset. We evaluate on the same models as in the input attribution tasks.

Baselines. We compare DPA against simple component-level baselines that rank internal units by their activations: *Attn-only*, which scores attention heads by head activation magnitude; *MLP-only*, which ranks neurons by activation strength; and *Norm*, which uses the L2-norm decomposed head or neuron contributions. We further evaluate on gradient-based methods (*Gradient at component*, and *AtP* (Nanda, 2023) as well as Information Flow Routes (IFR) (Ferrando & Voita, 2024).

Evaluation Setup. All attention heads and MLP neurons are ranked according to their attribution scores. We progressively ablate or recover the top-ranked components and measure the resulting change in the model's prediction confidence. Additional setup information for reproducibility are located in Appendix F.

Methods	Known 1000			SQuAD v2.0			IMDb		
	dis. ↓	rec. ↑	total ↑	dis. ↓	rec. ↑	total ↑	dis. ↓	rec. ↑	total ↑
Reference									
Random	14.22	22.78	8.56	50.64	52.39	1.75	84.44	85.93	1.49
Attention-based									
Last layer	**5.26**	24.83†	19.57 †	21.96	82.04	60.08	79.68†	91.57	11.89†
Mean	6.57	25.78	19.21	**17.06**	89.46†	72.40†	79.89	91.42	11.53
Rollout	13.26	15.72	2.46	40.58	69.43	28.85	85.33	92.47	7.14
Gradient-based									
Gradient	12.14	22.94	10.80	49.95	46.68	-3.27	84.52	84.58	0.06
Input X Gradient	12.58	21.27	8.69	34.07	63.22	29.15	82.65	84.34	1.69
Integrated Gradients	9.92	24.49	14.57	17.92	79.53	61.61	81.29	92.41†	11.12
Decomposition-based									
DePass	9.17	25.45	16.28	19.29	78.45	59.16	85.49	69.65	-15.84
Contextual Mixing-based									
IFR	16.15	11.88	-4.27	50.16	62.51	12.35	91.63	91.12	-0.51
Proposed Method									
Dual Path Attribution (DPA)	5.79†	**33.88**	**28.09**	17.73†	**92.51**	**74.78**	**69.34**	**99.51**	**30.17**

Table 1. Faithfulness evaluation of crucial token attribution on **Llama-3.1-8B-Instruct** across Known 1000 (factual knowledge), SQuAD v2.0 (reading comprehension), and IMDb (sentiment analysis). Lower disruption (*dis.*) and higher recovery (*rec.*) indicate more faithful token localization; *total* denotes their aggregate AUC score. **Bold** indicates the best result, and † the second best. DPA consistently achieves the highest faithfulness across datasets.

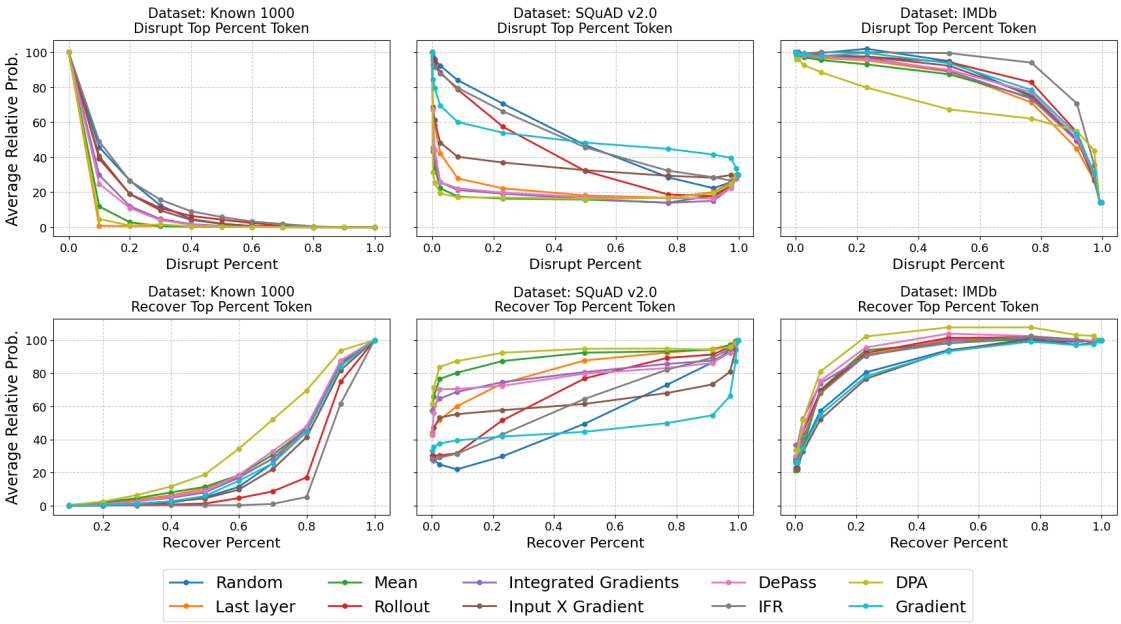

Figure 2. Performance comparison with baseline attribution methods on **Llama-3.1-8B-Instruct** under different top input token masking strategies. Our approach shows *lower* Top-k disruption and *higher* Top-k recovery, indicating more accurate identification of important components.

Results. As shown in Table 2 and Figure 3, DPA achieves zero disruption and high recovery on both IOI and Known 1000. Although Attn-only edges out DPA in aggregate AUC on IOI, DPA's S-shaped curve isolates the circuit far earlier (around 0.001%), pushing the target probability above 140%[3] before dipping as opposing nodes are reintroduced.

[3]Note that recovery can exceed 100% when ablation increases the target probability above that of the clean baseline.

Methods	Known 1000			IOI		
	dis. ↓	rec. ↑	total ↑	dis. ↓	rec. ↑	total ↑
Reference						
Random	19.77	27.55	7.78	21.85	27.54	5.69
Baselines						
Attn-only	0.66	93.43†	92.77†	0.15	**93.07**	**92.92**
MLP-only	0.37	27.58	27.21	0.87	27.53	26.66
Norm-only	0.06†	67.25	67.19	0.01†	65.02	65.01
Gradient-based						
Gradient	0.28	27.62	27.34	**0.00**	30.95	30.95
AtP	**0.00**	81.39	81.39	**0.00**	84.39	84.39
Contextual Mixing-based						
IFR	0.32	34.72	34.40	0.02	29.15	29.13
Proposed Method						
Dual Path Attribution (DPA)	**0.00**	**123.50**	**123.50**	**0.00**	86.87†	86.87†

Table 2. Component-level attribution performance on **Llama-3.1-8B-Instruct** across Known 1000 (factual knowledge) and IOI (Indirect Object Identification) benchmarks. Lower disruption (*dis.*) and higher recovery (*rec.*) indicate more faithful localization of causally relevant components; the *total* score aggregates both effects. **Bold** denotes the best result, and † indicates the second best. Our DPA consistently achieves the most faithful component localization across benchmarks.

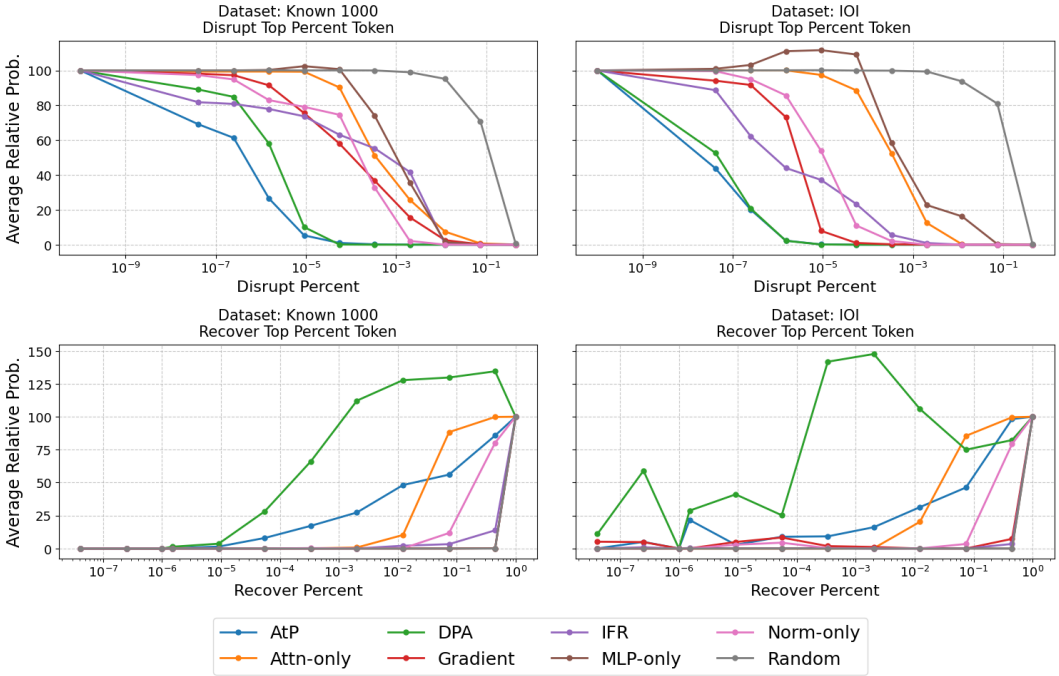

Figure 3. Performance comparison with baseline attribution methods on **Llama-3.1-8B-Instruct** under different top model component masking strategies. Our approach shows lower disruption and higher recovery, indicating more accurate identification of important components.

This dip, which lowers DPA's AUC, stems from (i) attributing logits rather than post-softmax probabilities—thus missing gains from inhibited competing logits—and (ii) approximation errors within IOI's highly specific circuit. The sharp early recovery nonetheless evidences DPA's precision in identifying causal mechanisms. Extended results for Llama2, Qwen3, Mistral, and Qwen2.5 appear in Appendix G.

6.2. Sensitivity Analysis

RQ 2: *Does the selection of balanced hyperparameters result in optimal performance?* In the sensitivity analysis, we evaluate how the attribution performance changes

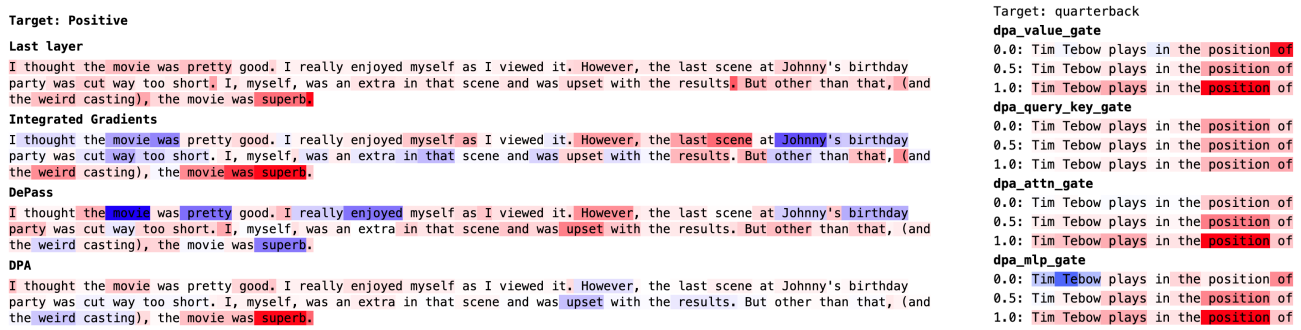

Figure 4. Qualitative attribution examples. Red highlights denote tokens that positively contribute to the target, while blue denotes tokens that hinder it. As shown on the IMDb dataset (left), DPA more precisely distinguishes between helpful and harmful context compared to baseline methods. The right panel illustrates DPA's attribution across different configurations (μ).

Sensitivity	μ_q	μ_k	μ_v	μ_{gate}	μ_{up}
Control-Content	$\frac{1-p}{2}$	$\frac{1-p}{2}$	p	$1-p$	p
Attention	$\frac{1-p}{2}$	$\frac{1-p}{2}$	p	0.5	0.5
Query-Key	$\frac{p}{2}$	$\frac{1-p}{2}$	0.5	0.5	0.5
MLP	0.25	0.25	0.5	$1-p$	p

Table 3. Hyperparameter configuration for the sensitivity analysis. $p \in [0, 1]$ is the sensitivity variable in our experiment.

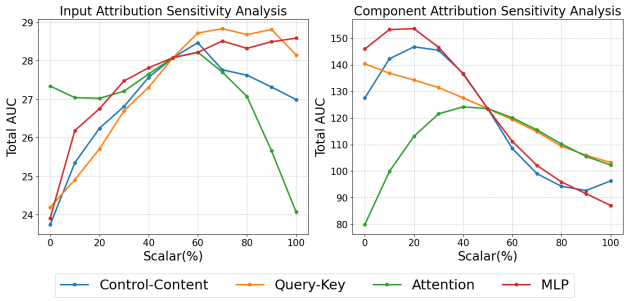

Figure 5. Sensitivity analysis of Dual Path Attribution (DPA) scaling configurations. We evaluate the impact of varying attribution parameters to prioritize different information pathways.

when altering the weighting parameters μ, which balance the contributions of different pathways. Table 3 shows the configuration weights. Despite the Query-Key sensitivity analysis, a higher p indicates larger reliance on the content pathways.

- **Control-Content Balance.** Varying p shifts emphasis from pure content ($p \approx 1$) to pure control ($p \approx 0$) across both modules, exposing how content–routing interactions affect faithfulness.
- **Query-Key Balance.** Within the attention control path, we fix $\mu_v = 0.5$ and redistribute the remaining weight between μ_q and μ_k to localize sensitivity to the 'request' (query) versus 'source' (key) of information.
- **Attention Balance.** With MLP weights fixed, we test whether attention-mediated effects are dominated by content being moved (μ_v) or routing logic (μ_q, μ_k).
- **MLP Balance.** We focus on the Gated Linear Unit by fixing the attention mechanism to the balanced configuration, shifting weight between μ_{gate} and μ_{up} to probe the gate's selective filtering behavior.

Results. Figures 4 and 5 summarize the qualitative and sensitivity results. Input attribution peaks in the mid-range (50–70%), indicating that macro-level token importance favors direct content pathways (e.g., value and up projections). In contrast, component attribution peaks at much smaller scales ($p \approx 0.2$) before sharp decline, revealing that fine-grained fidelity hinges on control/routing pathways (gates and queries/keys), with $1 - p \approx 0.8$ needed to capture individual functional roles. Balancing both, $p = 0.5$ offers the most stable trade-off and is used throughout this paper; we leave broader configuration sweeps to future work.

7. Limitations

While DPA offers an efficient framework for dense component attribution, several limitations remain. The decomposition is tailored to dense SwiGLU Transformers, so extending it to MoE or Mamba architectures requires deriving a new inverse function for nonlinearity. Its reliance on first-order Taylor expansions discards second-order cross-derivatives, degrading faithfulness in circuits with highly coupled or synergistic interactions. Attribution also targets the logit rather than post-softmax probabilities, so indirect mechanisms involving suppression of competing logits are not fully captured. Finally, despite $O(1)$ time complexity, caching dense intermediate activations remains memory-intensive on very large models or ultra-long contexts.

8. Conclusion

We introduced *Dual Path Attribution* (DPA), a mechanistic interpretability framework that bridges attribution faithful-

ness and computational efficiency. By decomposing the bilinear transformations of SwiGLU-based Transformers into complementary content and control pathways, DPA captures non-linear component interactions overlooked by linear approximations and enables exact, single-pass attribution with $O(1)$ component-level complexity. This isolates causal mechanisms down to individual attention heads and MLP neurons without the overhead of exhaustive activation patching. Extensive empirical evaluations confirm that DPA consistently achieves state-of-the-art faithfulness across diverse benchmarks while maintaining unprecedented computational efficiency, providing a robust and practical tool for uncovering the internal reasoning pathways of modern large language models.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grants (RS-2025-00513098) and (RS-2018-NR031059) as well as the BK21 FOUR program (41202420214871), both funded by the Ministry of Education of Korea. Access to the A100 GPU used in the experiments was provided by the Department of Data Convergence Computing at Kyungpook National University, where Young-Kyoon Suh serves as an adjunct professor.

References

Abnar, S. and Zuidema, W. Quantifying attention flow in transformers. In *Annual Meeting of the Association for Computational Linguistics*, 2020a. URL https://api.semanticscholar.org/CorpusID:218487351.

Abnar, S. and Zuidema, W. Quantifying attention flow in transformers. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.385. URL https://aclanthology.org/2020.acl-main.385/.

Achtibat, R., Hatefi, S. M. V., Dreyer, M., Jain, A., Wiegand, T., Lapuschkin, S., and Samek, W. Attnlrp: Attention-aware layer-wise relevance propagation for transformers, 2024. URL https://arxiv.org/abs/2402.05602.

Ali, A., Schnake, T., Eberle, O., Montavon, G., Müller, K.-R., and Wolf, L. Xai for transformers: Better explanations through conservative propagation. *ArXiv*, abs/2202.07304, 2022. URL https://api.semanticscholar.org/CorpusID:246863594.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10, 2015. URL https://api.semanticscholar.org/CorpusID:9327892.

Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question?, 2018. URL https://arxiv.org/abs/1702.08591.

Belrose, N., Furman, Z., Smith, L., Halawi, D., Ostrovsky, I. V., McKinney, L., Biderman, S., and Steinhardt, J. Eliciting latent predictions from transformers with the tuned lens. *ArXiv*, abs/2303.08112, 2023. URL https://api.semanticscholar.org/CorpusID:257504984.

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability. *ArXiv*, abs/2304.14997, 2023. URL https://api.semanticscholar.org/CorpusID:258418244.

Denil, M., Demiraj, A., and de Freitas, N. Extraction of salient sentences from labelled documents. *ArXiv*, abs/1412.6815, 2014. URL https://api.semanticscholar.org/CorpusID:9121062.

Denil, M., Demiraj, A., and de Freitas, N. Extraction of salient sentences from labelled documents, 2015. URL https://arxiv.org/abs/1412.6815.

DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R., and Wallace, B. C. ERASER: A benchmark to evaluate rationalized NLP models. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4443–4458, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL https://aclanthology.org/2020.acl-main.408/.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Ferrando, J. and Voita, E. Information flow routes: Automatically interpreting language models at scale. In

Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17432–17445, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.965. URL https://aclanthology.org/2024.emnlp-main.965/.

Ferrando, J., Gállego, G. I., and Costa-jussà, M. R. Measuring the mixing of contextual information in the transformer. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8698–8714, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.595. URL https://aclanthology.org/2022.emnlp-main.595/.

Ferrando, J., Gállego, G. I., Tsiamas, Y. I., and Costa-jussà, M. R. Explaining how transformers use context to build predictions. In *Annual Meeting of the Association for Computational Linguistics*, 2023. URL https://api.semanticscholar.org/CorpusID:258832652.

Ferrando, J., Sarti, G., Bisazza, A., and Costa-jussà, M. R. A primer on the inner workings of transformer-based language models. *ArXiv*, abs/2405.00208, 2024. URL https://api.semanticscholar.org/CorpusID:269484740.

Geva, M., Caciularu, A., Wang, K., and Goldberg, Y. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *ArXiv*, abs/2203.14680, 2022. URL https://api.semanticscholar.org/CorpusID:247762385.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz,

G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Hong, X., Jiang, C., Tian, K., Qi, B., Sun, Y., Ding, N., and Zhou, B. Depass: Unified feature attributing by simple decomposed forward pass. *ArXiv*, abs/2510.18462, 2025. URL https://api.semanticscholar.org/CorpusID:282246406.

Jafari, F. R., Eberle, O., Khakzar, A., and Nanda, N. RelP: Faithful and efficient circuit discovery in language models via relevance patching, 2025. URL http://arxiv.org/abs/2508.21258.

Jain, S. and Wallace, B. C. Attention is not explanation. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:67855860.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de Las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. URL https://api.semanticscholar.org/CorpusID:263830494.

Kobayashi, G., Kuribayashi, T., Yokoi, S., and Inui, K. Attention is not only a weight: Analyzing transformers with vector norms. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.574. URL https://aclanthology.org/2020.emnlp-main.574/.

Kobayashi, G., Kuribayashi, T., Yokoi, S., and Inui, K. Incorporating Residual and Normalization Layers into Analysis of Masked Language Models. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4547–4568, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.373. URL https://aclanthology.org/2021.emnlp-main.373/.

Kobayashi, G., Kuribayashi, T., Yokoi, S., and Inui, K. Analyzing feed-forward blocks in transformers through the lens of attention maps. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mYWsyTuiRp.

Li, J., Chen, X., Hovy, E. H., and Jurafsky, D. Visualizing and understanding neural models in nlp. In *North American Chapter of the Association for Computational Linguistics*, 2015. URL https://api.semanticscholar.org/CorpusID:14099741.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual*

Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt. In *Neural Information Processing Systems*, 2022. URL https://api.semanticscholar.org/CorpusID:255825985.

Modarressi, A., Fayyaz, M., Yaghoobzadeh, Y., and Pilehvar, M. T. GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V. (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 258–271, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.19. URL https://aclanthology.org/2022.naacl-main.19/.

Modarressi, A., Fayyaz, M., Aghazadeh, E., Yaghoobzadeh, Y., and Pilehvar, M. T. DecompX: Explaining transformers decisions by propagating token decomposition. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2649–2664, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.149. URL https://aclanthology.org/2023.acl-long.149/.

Nanda, N. Attribution patching: Activation patching at industrial scale. 2023. URL https://www.neelnanda.io/mechanistic-interpretability/attribution-patching.

nostalgebraist. Interpreting gpt: the logit lens. *AI Alignment Forum*, 2020. URL https://www.alignmentforum.org/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad. *ArXiv*, abs/1806.03822, 2018. URL https://api.semanticscholar.org/CorpusID:47018994.

Shazeer, N. Glu variants improve transformer. *ArXiv*, abs/2002.05202, 2020. URL https://api.semanticscholar.org/CorpusID:211096588.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences, 2019. URL https://arxiv.org/abs/1704.02685.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013. URL https://api.semanticscholar.org/CorpusID:1450294.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise, 2017. URL https://arxiv.org/abs/1706.03825.

Su, J., Lu, Y., Pan, S., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864, 2021. URL https://api.semanticscholar.org/CorpusID:233307138.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, 2017. URL https://api.semanticscholar.org/CorpusID:16747630.

Syed, A., Rager, C., and Conmy, A. Attribution patching outperforms automated circuit discovery. *ArXiv*, abs/2310.10348, 2023. URL https://api.semanticscholar.org/CorpusID:264147090.

Syed, A., Rager, C., and Conmy, A. Attribution patching outperforms automated circuit discovery. In Belinkov, Y., Kim, N., Jumelet, J., Mohebbi, H., Mueller, A., and Chen, H. (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 407–416, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.25. URL https://aclanthology.org/2024.blackboxnlp-1.25/.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y.,

Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., and Shieber, S. M. Investigating gender bias in language models using causal mediation analysis. In *Neural Information Processing Systems*, 2020. URL https://api.semanticscholar.org/CorpusID:227275068.

Voita, E., Sennrich, R., and Titov, I. Analyzing the source and target contributions to predictions in neural machine translation. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1126–1140, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.91. URL https://aclanthology.org/2021.acl-long.91/.

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *ArXiv*, abs/2211.00593, 2022. URL https://api.semanticscholar.org/CorpusID:253244237.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yang, Q. A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y.-C., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z., Quan, S., and Wang, Z. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024. URL https://api.semanticscholar.org/CorpusID:274859421.

Zhang, B. and Sennrich, R. Root mean square layer normalization. *ArXiv*, abs/1910.07467, 2019. URL https://api.semanticscholar.org/CorpusID:113405151.

Zhang, F. and Nanda, N. Towards best practices of activation patching in language models: Metrics and methods. *ArXiv*, abs/2309.16042, 2023. URL https://api.semanticscholar.org/CorpusID:263131114.

A. Transformer Architecture

We consider a decoder-only Transformer composed of a stack of L identical layers indexed by $l \in \{1, \ldots, L\}$. Each layer consists of two primary sub-modules: a *multi-head self-attention* (MHSA) block and a *gated feed-forward network* (GLU).

Given an input token sequence $t = (t_1, \ldots, t_T)$, tokens are mapped to vector embeddings $\mathbf{X}^{(0)} \in \mathbb{R}^{T \times d}$ using an embedding matrix $\mathbf{W}_E \in \mathbb{R}^{|\mathcal{V}| \times d}$, where $\mathcal{V}$ denotes the vocabulary. Representations propagate sequentially through the network layers. Under a pre-normalization configuration, the update rule for layer l is given by:

$$\mathbf{X}^{(l)}_{\text{mid}} = \mathbf{X}^{(l-1)} + \text{MHSA}\Big(\text{LN}(\mathbf{X}^{(l-1)})\Big),$$
$$\mathbf{X}^{(l)} = \mathbf{X}^{(l)}_{\text{mid}} + \text{GLU}\Big(\text{LN}(\mathbf{X}^{(l)}_{\text{mid}})\Big),$$

where $\text{LN}(\cdot)$ denotes the normalization applied prior to each sub-module. For notational convenience, we refer to normalized activations as $\tilde{\mathbf{X}}$ when needed.

The final representation $\mathbf{X}^{(L)}$ is projected into the vocabulary space via the unembedding matrix $\mathbf{W}_{UE} \in \mathbb{R}^{d \times |\mathcal{V}|}$, yielding the logits used for next-token prediction.

A key property enabling attribution analysis is the additive structure induced by residual connections around each sub-module, commonly referred to as the *residual stream*. This structure allows the final representation to be expressed as:

$$\mathbf{X}^{(L)} = \mathbf{X}^{(0)} + \sum_{l=1}^{L} \Big(\text{MHSA}(\tilde{\mathbf{X}}^{(l-1)}) + \text{GLU}(\tilde{\mathbf{X}}^{(l)}_{\text{mid}})\Big).$$

As a result, the contribution of any module output can be directly interpreted by projecting it into the vocabulary space via the unembedding matrix (Elhage et al., 2021; nostalgebraist, 2020).

A.1. Root-Mean-Squared Normalization (RMSNorm)

Root-Mean-Squared Normalization (RMSNorm) normalizes activations by their root-mean-square magnitude and is defined as:

$$\mathrm{RMSNorm}(\mathbf{X}) = \frac{\mathbf{X}}{\sqrt{\mathrm{Var}[\mathbf{X}] + \epsilon}} \odot \boldsymbol{\gamma}, \tag{1}$$

where ϵ is a small constant for numerical stability and $\boldsymbol{\gamma}$ denotes learnable scaling parameters.

A.2. Rotary Positional Embeddings (RoPE)

Rotary Positional Embeddings (RoPE) encode positional information by applying a rotation to query and key vectors. Given a vector $x \in \mathbb{R}^d$ at position m, RoPE defines the rotated representation as:

$$R(x, m) = \mathcal{R}_{\Theta,m} x, \tag{2}$$

where $\mathcal{R}_{\Theta,m} \in \mathbb{R}^{d \times d}$ is a block-diagonal rotation matrix composed of $d/2$ independent 2×2 rotation blocks:

$$\mathcal{R}_{\Theta,m} = \begin{pmatrix} M_1 & & 0 \\ & \ddots & \\ 0 & & M_{d/2} \end{pmatrix}. \tag{3}$$

Each block M_j rotates a pair of dimensions by an angle $m\theta_j$:

$$M_j = \begin{pmatrix} \cos(m\theta_j) & -\sin(m\theta_j) \\ \sin(m\theta_j) & \cos(m\theta_j) \end{pmatrix}. \tag{4}$$

The rotation frequencies are defined as a geometric progression:

$$\theta_j = 10000^{-2(j-1)/d}, \quad j \in \{1, \ldots, d/2\}. \tag{5}$$

An important property of RoPE is that the inner product between a query at position m and a key at position n depends only on their relative offset:

$$(R(q, m))^\top R(k, n) = q^\top \mathcal{R}_{\Theta,n-m} k. \tag{6}$$

B. Complexity Analysis

As large language models are increasingly deployed on tasks requiring long context windows, the computational and memory scalability of attribution methods becomes a critical bottleneck. In this section, we analyze the complexity of Dual Path Attribution (DPA) and compare it against two established baselines: Activation Patching (AP) and DePass (Hong et al., 2025). We evaluate these methods both theoretically and empirically, demonstrating that DPA uniquely decoupling attribution cost from the number of components, thereby enabling highly efficient dense attribution even over long sequences.

Methods	**Complexity** (w.r.t. M)	**Fwd passes**
Activation Patching (AP)	$O(M)$	M
DePass	$O(M)$	$\lceil M/B \rceil$
Dual Path Attribution (DPA)	**O(1)**	**0**

Table 4. Computational complexity comparison with respect to the number of attributed components M. Activation Patching and DePass require forward passes that scale linearly with M, whereas DPA performs attribution with no additional forward passes and a single backward pass.

B.1. Theoretical Complexity Analysis

As summarized in Table 4, we evaluate the computational complexity of Dual Path Attribution (DPA) against Activation Patching (AP) and DePass with respect to the number of attributed components M.

For a dense attribution task, AP requires M independent forward passes. Because each pass incurs the $O(S^2)$ time complexity of the attention mechanism (where S is the sequence length), its $O(M \cdot S^2)$ scaling becomes computationally prohibitive for long-context tasks. To mitigate this, methods like DePass leverage frozen activations through a modified forward pass, but still suffer from an $O(M)$ complexity bottleneck. DePass explicitly decomposes hidden representations into M independent contribution vectors. For large models (e.g., Llama-3.1-8B-Instruct with $N_{\text{mlp}} \approx 14{,}000$), processing these simultaneously is intractable, necessitating sequential batching that yields $\lceil M/B \rceil$ forward passes (where B is the batch size).

In contrast, DPA adopts a target-centric formulation that is completely independent of M. By propagating a single effective target vector backward from the output, DPA computes the attribution score for any component via a simple dot product with its cached contribution. Consequently, DPA requires 0 additional forward passes and only a single backward pass, achieving a rigorous $O(1)$ time complexity with respect to M. This decoupling effectively eliminates the prohibitive scaling overhead inherent in forward decomposition-based methods.

B.2. Empirical Measurements Analysis

We provide comprehensive empirical runtime and peak memory measurements across four distinct SwiGLU-based LLMs (representing the LLaMA, Qwen, and Mistral families) in Table 5. To ensure a strictly fair comparison, all evaluations were executed under identical hardware configurations, utilizing the same GPU setup and batch size. Furthermore, peak memory allocation was explicitly tracked using PyTorch's native CUDA memory allocator. This approach guarantees the precise measurement of tensor-level VRAM consumption directly managed by the model, effectively isolating the metric from OS-level caching overheads

or external system processes.

The empirical results demonstrate that our DPA framework significantly outperforms existing baseline methods in computational speed across all scenarios. Even on short-context datasets like Known 1000, DPA operates consistently faster than Activation Patching (AP) and DePass across all evaluated models. Furthermore, DPA exhibits exceptional scalability and efficiency on longer context lengths, such as the SQuAD v2.0 dataset. For instance, on the Llama-3.1-8B-Instruct model, DPA completed the attribution calculations in just 129.05 seconds. In stark contrast, both AP and DePass required approximately 1.5 hours (5370.65 and 5452.28 seconds, respectively) for the same task, representing a massive 40× speedup for DPA.

Crucially, DPA achieves these drastically faster runtimes while maintaining a peak memory footprint that is highly comparable to the baseline methods across all tested models. This confirms that DPA is a profoundly practical, robust, and resource-efficient solution for real-world tasks involving long context sequences, successfully overcoming the severe computational bottlenecks of traditional intervention-based methods.

C. Algorithm of Dual Path Attribution

The complete procedure for Dual Path Attribution (DPA) is formalized in Algorithm 1. The framework operates in a two-stage process: a single forward pass for activation caching followed by a recursive top-down target propagation. By decoupling the attribution cost from the number of attributed components M, DPA achieves $O(1)$ complexity, making it highly scalable for dense attribution tasks in large-scale transformers.

D. Derivations of DPA Target Tracing

In this section, we provide the complete mathematical derivations for the target propagation rules underlying Dual Path Attribution (DPA). By propagating a single effective target vector backward through the network, DPA analytically computes the gradient of the attribution score with respect to the intermediate representations. Below, we derive these exact target propagation formulas for the two primary computational blocks of the Transformer architecture: the Gated Linear Unit (GLU) and the Multi-Head Self-Attention (MHSA) module.

D.1. Gated Linear Unit (GLU) Target Propagation

Forward Pass Definition. Consider a single neuron n within the GLU layer. The contribution of this neuron to the residual stream is defined as:

$$\text{GLU}^{(n)}(\tilde{\mathbf{x}}) = \left(\sigma(\tilde{\mathbf{x}}\mathbf{w}_G^{(n)}) \odot (\tilde{\mathbf{x}}\mathbf{w}_U^{(n)})\right)\mathbf{w}_D^{(n)}$$

where:

- $\sigma(\cdot)$ is the SiLU (Swish) activation function.
- $\mathbf{w}_G^{(n)}, \mathbf{w}_U^{(n)}$ are the column vectors of the gate and up projection matrices.
- $\mathbf{w}_D^{(n)}$ is the row vector of the down projection matrix.
- $\odot$ denotes element-wise multiplication (scalar here for a single neuron).

The attribution of this neuron to a global target $\mathbf{t}$ is given by:

$$S^{(n)} = \text{GLU}^{(n)}(\tilde{\mathbf{x}}) \cdot \mathbf{t}$$

$$S^{(n)} = \underbrace{\sigma(\tilde{\mathbf{x}}\mathbf{w}_G^{(n)})}_{\text{activation } \alpha} \cdot \underbrace{(\tilde{\mathbf{x}}\mathbf{w}_U^{(n)})}_{\text{content } v} \cdot \underbrace{(\mathbf{w}_D^{(n)}\mathbf{t})}_{\text{scalar } \lambda}$$

D.2. Up-Projection (Value) Pathway

To identify the contribution of the information-carrying pathway, we linearize the operation by freezing the gate activation $\alpha^{(n)}$. The attribution becomes linear with respect to $\tilde{\mathbf{x}}$:

$$S_{up}^{(n)} \approx \alpha^{(n)} \cdot (\tilde{\mathbf{x}}\mathbf{w}_U^{(n)}) \cdot \lambda$$

The gradient of this score with respect to the input $\tilde{\mathbf{x}}$ (which represents the propagated target) is:

$$\mathbf{g}_{up}^{(n)}(\mathbf{t}) = \nabla_{\tilde{\mathbf{x}}} S_{up}^{(n)} = \alpha^{(n)}\lambda\mathbf{w}_U^{(n)}$$

Incorporating the RMSNorm scaling parameters $\boldsymbol{\gamma}$ into the effective weight $\tilde{\mathbf{w}}$, we get the paper's formulation:

$$\boxed{\mathbf{g}_{up}^{(n)}(\mathbf{t}) = (\tilde{\mathbf{w}}_U^{(n)}) \cdot \alpha^{(n)} \cdot (\mathbf{w}_D^{(n)}\mathbf{t})}$$

D.3. Gate-Projection Pathway

For the gating pathway, we freeze the content term $v = \tilde{\mathbf{x}}\mathbf{w}_U^{(n)}$ and attribute through the non-linear activation $\sigma(s)$, where $s = \tilde{\mathbf{x}}\mathbf{w}_G^{(n)}$.

$$S_{gate}^{(n)} \approx \sigma(s) \cdot v \cdot \lambda$$

Using the linear approximation $\sigma(s) \approx s \cdot \frac{\sigma(s)}{s} = s \cdot \frac{\alpha^{(n)}}{s}$, we rewrite the score:

$$S_{gate}^{(n)} \approx (\tilde{\mathbf{x}}\mathbf{w}_G^{(n)}) \cdot \frac{\alpha^{(n)}}{s} \cdot v \cdot \lambda$$

Taking the gradient with respect to $\tilde{\mathbf{x}}$:

$$\mathbf{g}_{gate}^{(n)}(\mathbf{t}) = \nabla_{\tilde{\mathbf{x}}} S_{gate}^{(n)} = \mathbf{w}_G^{(n)} \cdot \frac{\alpha^{(n)}}{s} \cdot v \cdot \lambda$$

Grouping terms yields the final propagation formula:

$$\boxed{\mathbf{g}_{gate}^{(n)}(\mathbf{t}) = (\tilde{\mathbf{w}}_G^{(n)}) \cdot \frac{\alpha^{(n)}}{s} \cdot \left(v \cdot \mathbf{w}_D^{(n)}\mathbf{t}\right)}$$

Methods	Known 1000		SQuAD v2.0	
	Runtime (s)↓	Peak Memory (MB)↓	Runtime (s)↓	Peak Memory (MB)↓
Llama-3.1-8B-Instruct				
Activation Patching (AP)	14.88	20983.48	5370.65	15649.38
DePass	14.91	20663.91	5452.28	17210.68
Dual Path Attribution (DPA)	**4.98**	21603.86	**129.05**	17512.21
Llama-2-7B-Chat				
Activation Patching (AP)	17.58	24642.69	7229.39	13505.32
DePass	15.74	17321.58	6987.46	14976.61
Dual Path Attribution (DPA)	**5.33**	19449.77	**141.58**	15026.57
Qwen3-4B-Instruct-2507				
Activation Patching (AP)	10.32	15354.79	7336.73	7863.08
DePass	9.04	13866.60	4905.13	8591.08
Dual Path Attribution (DPA)	**4.00**	13985.03	**119.43**	9571.57
Mistral-7B-Instruct-v0.3				
Activation Patching (AP)	17.33	18395.77	9603.84	13973.31
DePass	16.87	19795.21	10808.99	14814.81
Dual Path Attribution (DPA)	**5.55**	21182.21	**170.92**	16191.69

Table 5. Empirical runtime and peak memory measurements across various SwiGLU-based LLMs (representing the LLaMA, Qwen, and Mistral families) on datasets with different context lengths. Notably, DPA achieves vastly superior computational efficiency, delivering drastically faster runtimes while maintaining a comparable peak memory footprint to the baseline methods. It maintains highly robust performance even on the long-context SQuAD v2.0 dataset, successfully overcoming the severe computational bottlenecks of AP and DePass.

D.4. Multi-Head Self-Attention (MHSA) Target Propagation

Forward Pass Definition. The output of an attention head h at query position i is:

$$\mathbf{y}_i^{(h)} = \sum_{j \le i} \alpha_{ij} (\tilde{\mathbf{x}}_j \mathbf{W}_V^{(h)}) \mathbf{W}_O^{(h)}$$

where the attention weights α_{ij} are computed via Softmax over the scaled dot-product of Rotary Positional Embedding (RoPE) transformed queries and keys:

$$\alpha_{ij} = \text{Softmax}_j(s_{ij})$$

$$s_{ij} = \frac{R_Q(\tilde{\mathbf{x}}_i \mathbf{W}_Q^{(h)}) \cdot R_K(\tilde{\mathbf{x}}_j \mathbf{W}_K^{(h)})}{\sqrt{d_h}}$$

The attribution to the target $\mathbf{t}$ is $S = \mathbf{y}_i^{(h)} \cdot \mathbf{t}$.

Value Pathway. We assume the attention pattern α_{ij} is fixed. The operation becomes linear with respect to the input $\tilde{\mathbf{x}}_j$ via the value matrix $\mathbf{W}_V$:

$$S_{value} = \sum_{j \le i} \alpha_{ij} (\tilde{\mathbf{x}}_j \mathbf{W}_V^{(h)}) (\mathbf{W}_O^{(h)} \mathbf{t})$$

The propagated target to token j is derived by taking the gradient w.r.t $\tilde{\mathbf{x}}_j$:

$$\boxed{\mathbf{g}_{V,j}^{(h)}(\mathbf{t}) = \tilde{\mathbf{W}}_V^{(h)} \left(\sum_{i \ge j} \alpha_{ij} (\mathbf{W}_O^{(h)} \mathbf{t}) \right)}$$

Query and Key Pathways (Softmax Decomposition). To propagate through the Query and Key paths, we must differentiate the Softmax function. Let the projected target vector be $\mathbf{t}' = \mathbf{W}_O^{(h)} \mathbf{t}$ and the value vector be $\mathbf{v}_j = \tilde{\mathbf{x}}_j \mathbf{W}_V^{(h)}$. The attribution score is:

$$S = \sum_k \alpha_{ik} (\mathbf{v}_k \cdot \mathbf{t}')$$

We first find the sensitivity of S with respect to the pre-softmax logits s_{ij}. Using the derivative of Softmax, $\frac{\partial \alpha_{ik}}{\partial s_{ij}} =$

Algorithm 1 Dual Path Attribution (DPA)

Require: Input tokens x, target logit direction $t^{(L)}$, sensitivity parameter p
Ensure: Attribution scores S_t for all components

Stage 1: Forward Pass and Activation Caching
Execute a single forward pass through L layers.
for $l = 1$ to L **do**
  Cache attention weights $\alpha^{(h)}$ and normalized inputs $\tilde{x}^{(l)}$.
  Cache GLU pre-activations $s^{(n)}$ and gate activations $\alpha^{(n)}$.
end for
Stage 2: Top-Down Target Propagation
Initialize effective target $t^{(L)} \leftarrow W_U^\top(t)$.
for $l = L$ down to 1 **do**
  Propagate through the Gated Linear Unit.
  Compute $g_{\text{up}}^{(n)}(t)$ and $g_{\text{gate}}^{(n)}(t)$ using linearized inverse rules.
  $t_{\text{mid}}^{(l)} \leftarrow t^{(l)} + \sum_n \left(\mu_{\text{up}} g_{\text{up}}^{(n)}(t^{(l)}) + \mu_{\text{gate}} g_{\text{gate}}^{(n)}(t^{(l)})\right)$.
  Propagate through Multi-Head Self-Attention.
  for each head h **do**
    Compute value path $g_V^{(h)}(t_{\text{mid}}^{(l)})$ and control paths $g_Q^{(h)}, g_K^{(h)}$.
    $g_{\text{head}}^{(h)} \leftarrow \mu_Q g_Q^{(h)} + \mu_K g_K^{(h)} + \mu_V g_V^{(h)}$.
  end for
  $t^{(l-1)} \leftarrow t_{\text{mid}}^{(l)} + \sum_h g_{\text{head}}^{(h)}$.
end for
Stage 3: Score Calculation
$S_t(x_i^{(0)}) \leftarrow x_i^{(0)} \cdot t_i^{(0)}$ *Input token attribution*
$S_t(h^{(l)}) \leftarrow \text{Attn}^{(h)}(\tilde{x}^{(l-1)}) \cdot t_{\text{mid}}^{(l)}$ *Attention head attribution*
$S_t(n^{(l)}) \leftarrow \text{GLU}^{(n)}(\tilde{x}_{\text{mid}}^{(l)}) \cdot t^{(l)}$ *MLP neuron attribution*
return S_t

$\alpha_{ij}(\delta_{jk} - \alpha_{ik})$:

$$\begin{aligned}\frac{\partial S}{\partial s_{ij}} &= \sum_k \frac{\partial \alpha_{ik}}{\partial s_{ij}} (\mathbf{v}_k \cdot \mathbf{t}') \\ &= \sum_k \alpha_{ij}(\delta_{jk} - \alpha_{ik})(\mathbf{v}_k \cdot \mathbf{t}') \\ &= \alpha_{ij}\left(\mathbf{v}_j \cdot \mathbf{t}' - \left(\sum_k \alpha_{ik}\mathbf{v}_k\right) \cdot \mathbf{t}'\right)\end{aligned}$$

Defining the expected head output as $\boldsymbol{\mu}_i^{(h)} = \sum_k \alpha_{ik}\mathbf{v}_k$, we obtain the scalar interaction term δ_{ij}:

$$\delta_{ij} = \alpha_{ij}\big(\mathbf{v}_j - \boldsymbol{\mu}_i^{(h)}\big) \cdot (\mathbf{W}_O^{(h)}\mathbf{t})$$

Now we propagate δ_{ij} through the dot product s_{ij}.

Query Path. The gradient of s_{ij} with respect to the query input vector $\mathbf{q}_i = \tilde{\mathbf{x}}_i \mathbf{W}_Q$ is proportional to the key vector $\mathbf{k}_j$.

$$\nabla_{\tilde{\mathbf{x}}_i} s_{ij} = \mathbf{W}_Q^{(h)} R_{Q,i}^{-1}\left(\frac{R_K(\tilde{\mathbf{x}}_j \mathbf{W}_K^{(h)})}{\sqrt{d_h}}\right)$$

Aggregating over all j, the target for the query input at position i is:

$$\boxed{\mathbf{g}_{Q,i}^{(h)}(\mathbf{t}) = \tilde{\mathbf{W}}_Q^{(h)} \sum_{j \le i} \delta_{ij} \cdot R_{Q,i}^{-1}\left(\frac{R_K(\tilde{\mathbf{x}}_j \mathbf{W}_K^{(h)})}{\sqrt{d_h}}\right)}$$

Key Path. Similarly, the gradient of s_{ij} with respect to the key input vector $\mathbf{k}_j = \tilde{\mathbf{x}}_j \mathbf{W}_K$ is proportional to the query vector $\mathbf{q}_i$.

$$\nabla_{\tilde{\mathbf{x}}_j} s_{ij} = \mathbf{W}_K^{(h)} R_{K,j}^{-1}\left(\frac{R_Q(\tilde{\mathbf{x}}_i \mathbf{W}_Q^{(h)})}{\sqrt{d_h}}\right)$$

Aggregating over all i that attend to j (i.e., $i \ge j$ in causal attention), the target for the key input at position j is:

$$\boxed{\mathbf{g}_{K,j}^{(h)}(\mathbf{t}) = \tilde{\mathbf{W}}_K^{(h)} \sum_{i \ge j} \delta_{ij} \cdot R_{K,j}^{-1}\left(\frac{R_Q(\tilde{\mathbf{x}}_i \mathbf{W}_Q^{(h)})}{\sqrt{d_h}}\right)}$$

E. Dataset Configurations

In this section, we provide detailed information regarding the licenses, usage, and preprocessing steps for the datasets

used to evaluate the Dual Path Attribution (DPA) framework. We conduct experiments across four representative tasks: factual knowledge, reading comprehension, sentiment analysis, and circuit discovery.

E.1. Licenses and Usage

All datasets utilized in this study are publicly accessible and were used in strict accordance with their respective licensing terms for research purposes.

- **Known 1000.** This dataset is distributed under the MIT License. It consists of 1,000 factual statements designed to evaluate how language models represent and utilize factual associations.
- **SQuAD v2.0.** The Stanford Question Answering Dataset is licensed under Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0). It is a standard benchmark for reading comprehension, requiring the model to identify answer spans within a given context or determine if a question is unanswerable.
- **IMDb Sentiment.** Usage of this dataset follows the IMDb Non-Commercial Licensing terms, which allow for limited personal and non-commercial use. In compliance with these terms, we acknowledge that the information is courtesy of IMDb (https://www.imdb.com) and is used with permission. The data was sourced from official contributor datasets to ensure adherence to copyright and conditions of use. It contains movie reviews labeled for binary sentiment analysis.
- **Indirect Object Identification (IOI).** This dataset is distributed under the MIT License. It serves as a specialized benchmark for component-level circuit discovery.

E.2. Dataset Preprocessing Details

To maintain a consistent evaluation signal and manage computational complexity, we performed systematic preprocessing as follows:

- **Known 1000.** We specifically selected instances where the model provided a correct baseline prediction for the target entity to ensure that the attribution was grounded in existing model knowledge. The data was then formatted as a prefix-completion task, ensuring that target entities consisted of short token spans to facilitate precise attribution measurement.
- **SQuAD v2.0.** Our processing focused exclusively on answerable questions to evaluate the model's ability to attribute information to the provided context. The preprocessing involved normalizing whitespace and special characters while ensuring all questions concluded with a proper question mark. Furthermore, we utilized the answer_start index to establish exact ground-truth tokens and filtered for answers consisting of no more than two words to ensure the localization signal remained clear and unambiguous.
- **IMDb Sentiment.** We sampled a balanced subset of 1,000 reviews, comprising 500 positive and 500 negative samples to prevent sentiment bias. The text was cleaned by removing HTML tags and collapsing redundant spaces to prevent tokenization artifacts. To ensure sufficient context for attribution while maintaining computational efficiency, we filtered reviews to maintain a character length between 200 and 500.
- **Indirect Object Identification (IOI).** This dataset was constructed using fixed linguistic templates to isolate the specific circuit responsible for identifying indirect objects. Our preprocessing focused on identifying the precise token positions of the subjects and indirect objects within these templates, which allowed us to measure the direct and indirect effects of internal components with high resolution.

F. Experiment Reproducibility

All experiments, including model inference and the recursive propagation of Dual Path Attribution (DPA) targets, were conducted on a high-performance computing cluster to ensure a consistent and high-throughput execution environment. The specific hardware and software configurations used for our empirical evaluations are as follows :

- **Hardware Resources.** We utilized an NVIDIA DGX system equipped with NVIDIA A100-SXM4-40GB GPUs. Each GPU provides 40 GB of HBM2 memory, which is essential for caching the high-dimensional activations required for attribution across large-scale models like Llama-3-8B-Instruct. The high memory bandwidth of the A100 architecture facilitated efficient processing of dense attribution tasks.
- **Software Environment.** The system operated on CUDA Version 12.2 with NVIDIA Driver Version 535.161.08. This environment supports the latest optimization libraries required for executing transformer-based models and performing efficient tensor operations during the target propagation stage.
- **Implementation Details.** The DPA framework and all comparative baselines, such as Integrated Gradients and DePass, were implemented using the PyTorch

library. Attribution results across factual knowledge, reading comprehension, and sentiment analysis tasks were obtained through a single forward pass for activation caching followed by a single top-down backward pass.

G. Extended Results

In this section, we evaluate the generalizability of Dual Path Attribution (DPA) across different model architectures to ensure our findings are robust and not strictly tied to a single model family.

G.1. Generalization within the LLaMA Family

We initially extended our experiments to Llama-2-7B-Chat not only Llama-3.1-8B-Instruct. The empirical results demonstrate that DPA maintains consistent attribution accuracy and computational efficiency within the LLaMA architecture.

The input and component attribution results for Llama-2-7B-Chat are shown in Table 6, and 7 (with corresponding Figures 6 and 7).

G.2. Generalization to Non-LLaMA SwiGLU Architectures

To address concerns regarding generalization to other model families, we conducted additional experiments on the Qwen and Mistral model families, specifically Qwen3-4B-Instruct-2507, Qwen2.5-32B-Instruct, and Mistral-7B-Instruct-v0.3. While these models share a SwiGLU-based Transformer backbone with the LLaMA series, they are fundamentally distinct. They feature structural variations in the attention mechanism (e.g., Sliding Window Attention in Mistral), utilize different tokenizers, and undergo entirely different pre- and post-training pipelines.

We present quantitative results and visualizations for Qwen3-4B-Instruct-2507 in Tables 8 and 9 (with corresponding Figures 8 and 9). Similarly, results for Qwen2.5-32B-Instruct are detailed in Tables 10 and 11 (Figures 10 and 11), and for Mistral-7B-Instruct-v0.3 in Tables 12 and 13 (Figures 12 and 13). Due to compute constraint results for Qwen2.5-32B-Instruct are constraint to the Known1000 dataset.

Empirical results demonstrate that DPA exhibits robust superiority over existing baseline methods across these architectural variations. By consistently achieving state-of-the-art performance in the majority of evaluations, DPA confirms its broad applicability and the stability of its performance.

Methods	Known 1000			SQuAD v2.0			IMDb		
	dis. ↓	rec. ↑	total ↑	dis. ↓	rec. ↑	total ↑	dis. ↓	rec. ↑	total ↑
Reference									
Random	13.17	20.64	7.47	23.74	25.04	1.30	58.98	57.80	-1.18
Attention-based									
Last layer	6.53	28.85†	22.32†	8.90	31.95	23.05	59.81	60.94	1.13
Mean	5.97†	25.10	19.13	5.48†	33.35†	27.87†	**46.75**	**78.60**	**31.85**
Rollout	15.45	16.17	0.72	22.64	27.35	4.71	83.31	55.54	-27.77
Gradient-based									
Gradient	12.14	18.75	6.61	16.94	15.55	-1.39	55.16	63.90	8.74
Input×Gradient	11.31	20.31	9.00	14.73	17.16	2.43	69.39	50.85	-18.54
Integrated Gradients	11.58	25.30	13.72	9.87	26.08	16.21	52.30	68.19	15.89
Decomposition-based									
DePass	7.55	26.29	18.74	12.62	27.49	14.87	68.19	56.13	-12.06
Contextual mixing-based									
IFR	15.22	12.76	-2.46	12.62	26.29	2.65	91.62	48.67	-42.95
Proposed Method									
Dual Path Attribution (DPA)	**5.60**	**34.41**	**28.81**	**3.73**	**36.08**	**32.35**	48.21†	74.53†	26.32†

Table 6. Faithfulness evaluation of crucial token attribution on **Llama-2-7B-Chat** across **Known 1000** (factual knowledge), **SQuAD v2.0** (reading comprehension), and **IMDb** (sentiment analysis). Lower disruption (*dis.*) and higher recovery (*rec.*) indicate more faithful token localization; *total* denotes their aggregate AUC score. **Bold** indicates the best result, and † the second best. Dual Path Attribution consistently achieves the highest faithfulness across datasets.

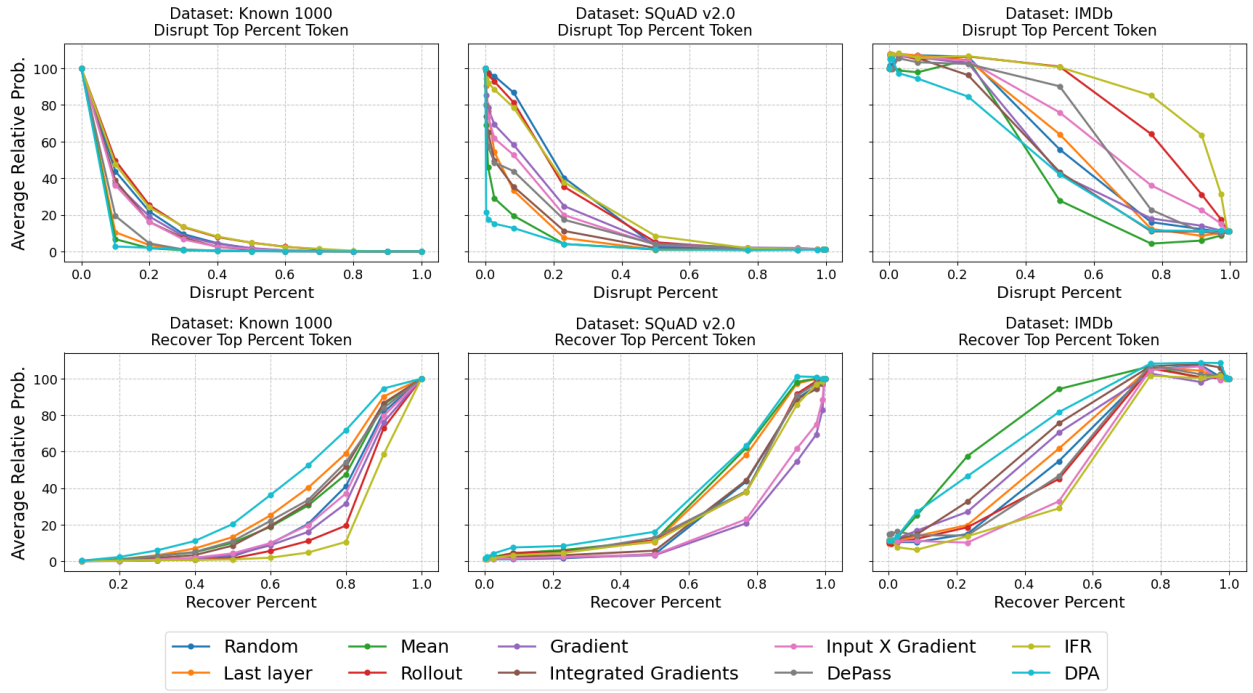

Figure 6. Performance comparison with baseline attribution methods on **Llama-2-7B-Chat** under different top input token masking strategies. Our approach shows *lower* Top-k disruption and *higher* Top-k recovery, indicating more accurate identification of important components.

Methods	Known 1000			IOI		
	dis. ↓	rec. ↑	total ↑	dis. ↓	rec. ↑	total ↑
Reference						
Random	22.89	27.65	4.76	24.07	27.67	3.60
Baselines						
Attn-only	2.34	92.58	90.24	0.53	91.39†	90.86†
MLP-only	1.74	29.16	27.42	5.59	27.58	21.99
Norm-only	0.24	64.68	64.44	0.05	66.98	66.93
Gradient-based						
Gradient	0.22†	28.21	27.99	0.01†	30.69	30.68
AtP	**0.00**	120.80†	120.80†	**0.00**	76.30	76.30
Contextual mixing-based						
IFR	0.69	30.49	29.80	0.48	28.86	28.38
Proposed Method						
Dual Path Attribution (DPA)	**0.00**	**136.45**	**136.45**	**0.00**	**127.16**	**127.16**

Table 7. Component-level attribution performance on **Llama-2-7B-Chat** across **Known 1000** (factual knowledge) and **IOI** (Indirect Object Identification) benchmarks. Lower disruption (*dis.*) and higher recovery (*rec.*) indicate more faithful localization of causally relevant components; the *total* score aggregates both effects. **Bold** denotes the best result, and † indicates the second best. Dual Path Attribution consistently achieves the most faithful component localization across benchmarks.

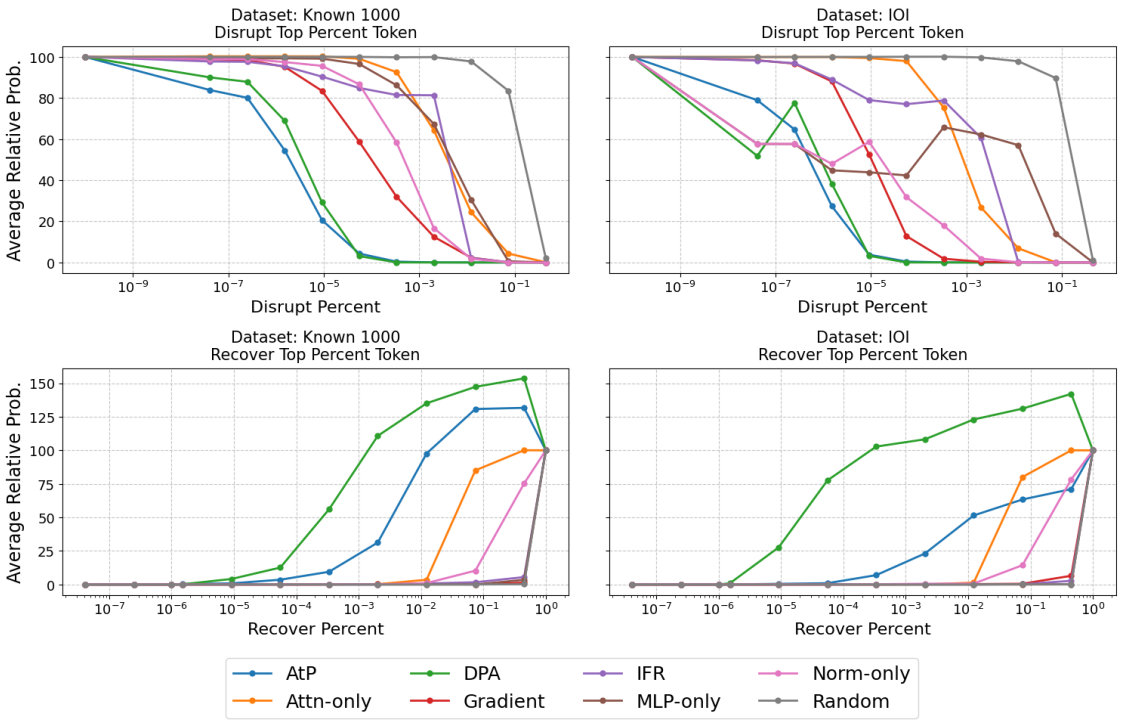

Figure 7. Performance comparison with baseline attribution methods on **Llama-2-7B-Chat** under different top model component masking strategies. Our approach shows lower Top-k disruption and higher Top-k recovery, indicating more accurate identification of important components.

Methods	Known 1000			SQuAD v2.0		
	dis. ↓	rec. ↑	total ↑	dis. ↓	rec. ↑	total ↑
Reference						
Random	11.52	19.64	8.12	48.89	50.00	1.11
Attention-based						
Last layer	8.03	22.82†	14.79	26.04	72.41	46.37
Mean	9.50	22.35	12.85	**11.69**	91.55†	79.86†
Rollout	15.25	11.79	-3.46	43.81	66.12	22.31
Gradient-based						
Gradient	10.88	18.07	7.19	57.82	35.49	-22.33
Input×Gradient	10.54	17.95	7.41	46.23	47.40	1.17
Integrated Gradients	7.55	16.18	8.63	41.39	49.99	8.60
Decomposition-based						
DePass	**6.24**	**26.74**	**20.50**	13.74	66.19	52.45
Contextual mixing-based						
IFR	15.80	11.28	-4.52	43.74	66.46	22.72
Proposed Method						
Dual Path Attribution (DPA)	6.54†	22.37	15.83†	11.93†	**93.86**	**81.93**

Table 8. Faithfulness evaluation of crucial token attribution on **Qwen3-4B-Instruct-2507** across **Known 1000** (factual knowledge), and **SQuAD v2.0** (reading comprehension). Lower disruption (*dis.*) and higher recovery (*rec.*) indicate more faithful token localization; *total* denotes their aggregate AUC score. **Bold** indicates the best result, and † the second best. Dual Path Attribution consistently achieves the highest faithfulness across datasets.

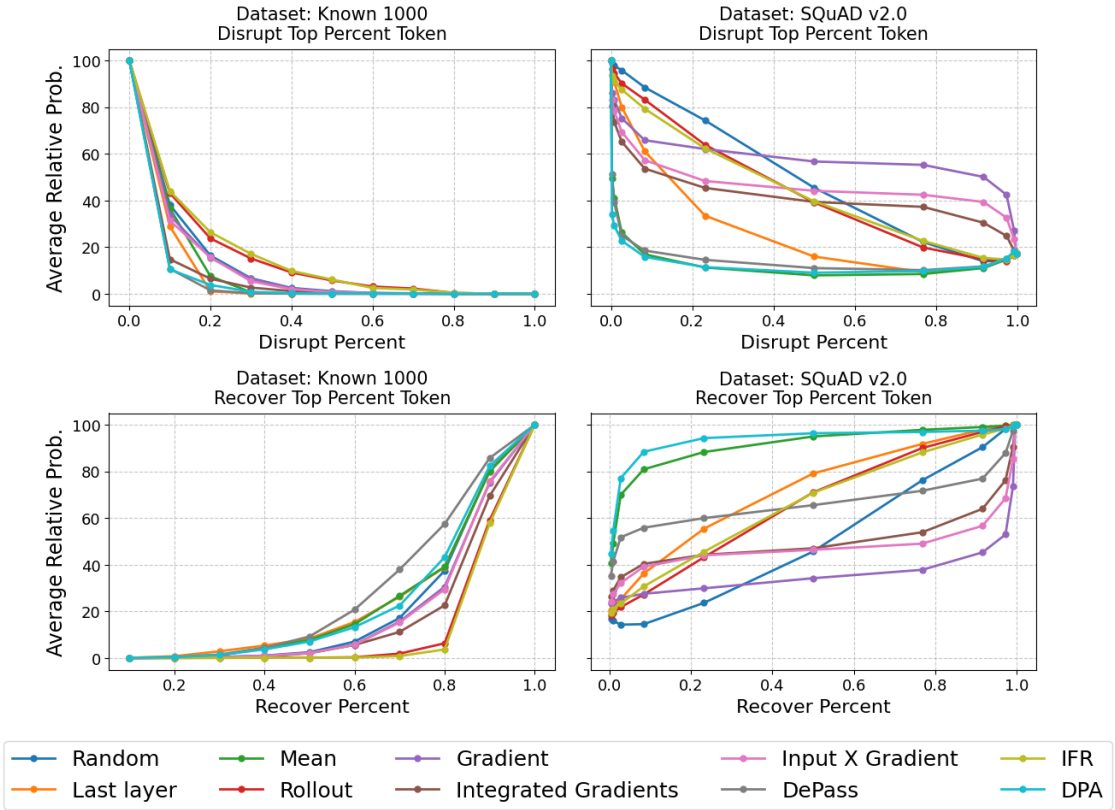

Figure 8. Performance comparison with baseline attribution methods on **Qwen3-4B-Instruct-2507** under different top input token masking strategies. Our approach shows *lower* Top-k disruption and *higher* Top-k recovery, indicating more accurate identification of important components.

Methods	Known 1000			IOI		
	dis. ↓	rec. ↑	total ↑	dis. ↓	rec. ↑	total ↑
Reference						
Random	19.00	27.53	8.53	22.63	27.55	4.92
Baselines						
Attn-only	2.13	74.12†	71.99†	0.66	**83.11**	**82.45**
MLP-only	0.01†	28.66	28.65	0.01†	27.56	27.55
Norm-only	**0.00**	47.64	47.64	**0.00**	28.68	28.68
Gradient-based						
Gradient	0.03	27.54	27.51	**0.00**	29.54	29.54
AtP	**0.00**	41.98	41.98	**0.00**	32.25	32.25
Contextual mixing-based						
IFR	0.17	38.68	38.51	0.02	28.59	28.57
Proposed Method						
Dual Path Attribution (DPA)	**0.00**	**105.21**	**104.35**	**0.00**	54.91†	54.91†

Table 9. Component-level attribution performance on **Qwen3-4B-Instruct-2507** across **Known 1000** (factual knowledge) and **IOI** (Indirect Object Identification) benchmarks. Lower disruption (*dis.*) and higher recovery (*rec.*) indicate more faithful localization of causally relevant components; the *total* score aggregates both effects. **Bold** denotes the best result, and † indicates the second best. Dual Path Attribution consistently achieves the most faithful component localization across benchmarks.

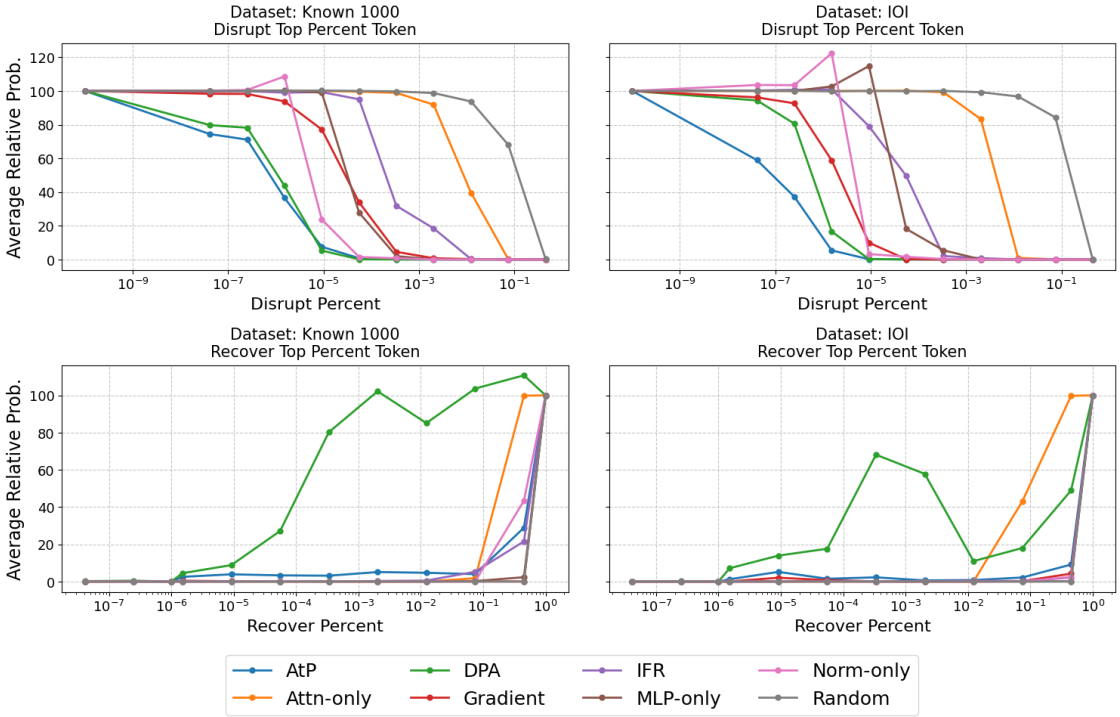

Figure 9. Performance comparison with baseline attribution methods on **Qwen3-4B-Instruct-2507** under different top model component masking strategies. Our approach shows lower Top-k disruption and higher Top-k recovery, indicating more accurate identification of important components.

Methods	Known 1000		
	dis. ↓	rec. ↑	total ↑
Reference			
Random	20.00	29.79	9.79
Attention-based			
Last layer	10.63†	32.18	21.55†
Mean	12.06	33.07†	21.01
Rollout	19.70	11.37	-8.33
Gradient-based			
Gradient	19.44	26.67	7.23
Input×Gradient	21.52	26.92	5.40
Integrated Gradients	18.60	27.70	9.10
Decomposition-based			
DePass	12.29	20.53	8.24
Contextual mixing-based			
IFR	18.75	26.61	7.86
Proposed Method			
Dual Path Attribution (DPA)	**10.32**	**41.35**	**31.03**

Table 10. Faithfulness evaluation of crucial token attribution for **Qwen2.5-32B-Instruct** on **Known 1000** (factual knowledge). Lower disruption (*dis.*) and higher recovery (*rec.*) indicate more faithful token localization; *total* denotes their aggregate AUC score. **Bold** indicates the best result, and † the second best. Dual Path Attribution consistently achieves the highest faithfulness across datasets.

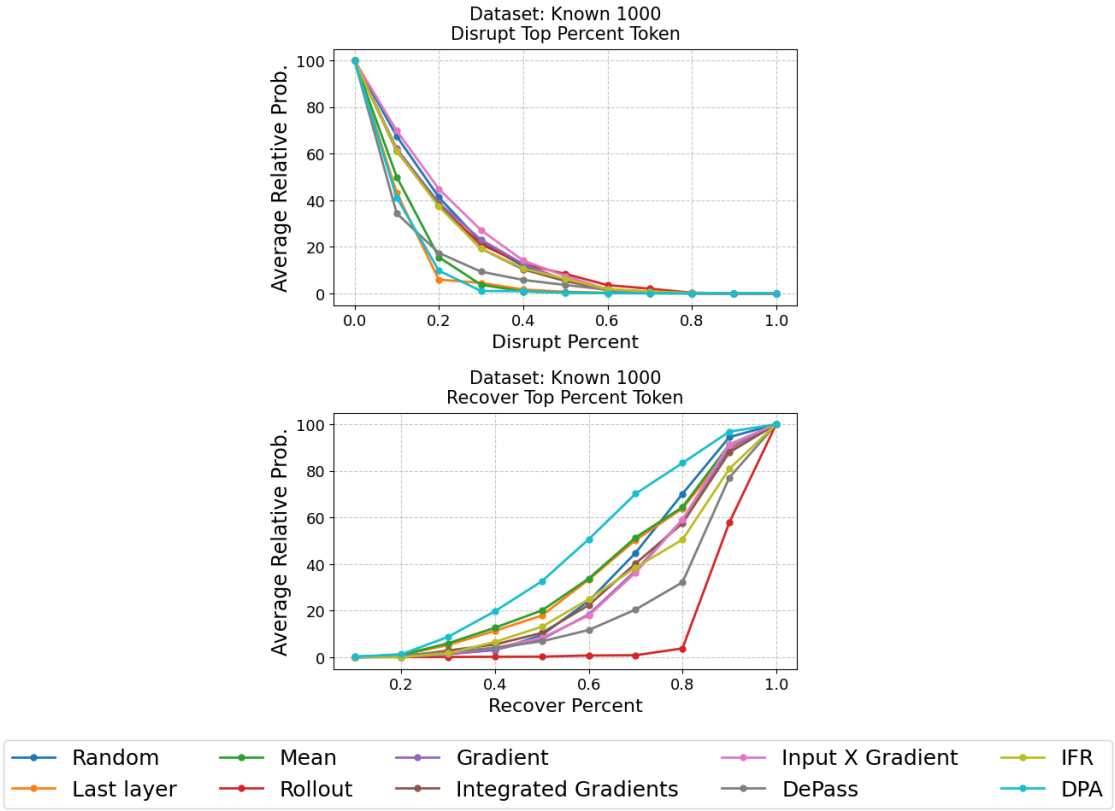

Figure 10. Performance comparison with baseline attribution methods on **Qwen2.5-32B-Instruct** under different top input token masking strategies. Our approach shows *lower* Top-k disruption and *higher* Top-k recovery, indicating more accurate identification of important components.

Methods	Known 1000		
	dis. ↓	rec. ↑	total ↑
Reference			
Random	22.59	27.55	4.96
Baselines			
Attn-only	4.25	74.29	70.04
MLP-only	0.01†	47.25	47.24
Norm-only	**0.00**	67.75†	67.75†
Gradient-based			
Gradient	0.05	27.56	27.51
AtP	**0.00**	51.69	51.69
Contextual mixing-based			
IFR	0.38	31.29	30.91
Proposed Method			
Dual Path Attribution (DPA)	**0.00**	**187.56**	**187.56**

Table 11. Component-level attribution performance for **Qwen2.5-32B-Instruct** on **Known 1000** (factual knowledge) benchmark. Lower disruption (*dis.*) and higher recovery (*rec.*) indicate more faithful localization of causally relevant components; the *total* score aggregates both effects. **Bold** denotes the best result, and † indicates the second best. Dual Path Attribution consistently achieves the most faithful component localization across benchmarks.

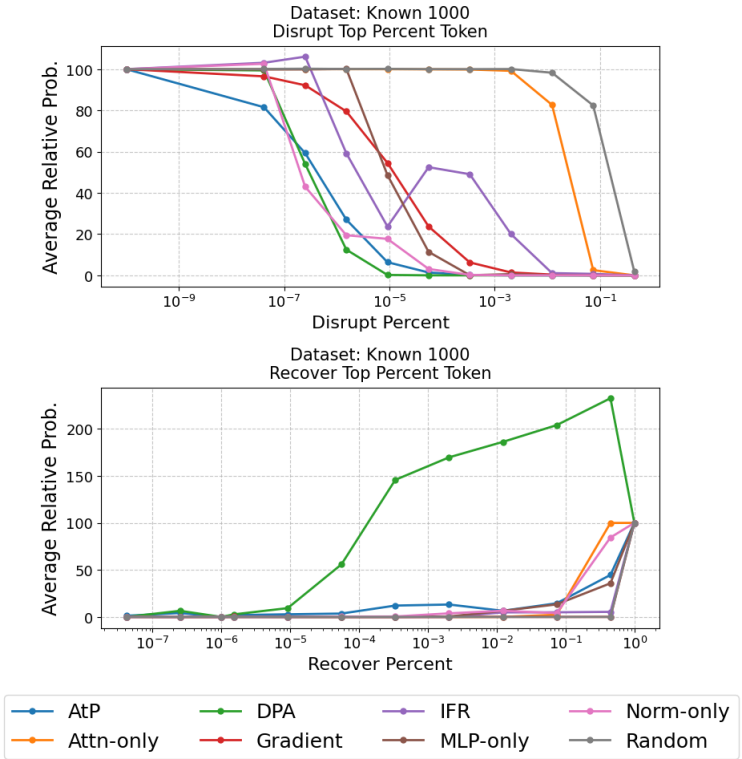

Figure 11. Performance comparison with baseline attribution methods on **Qwen2.5-32B-Instruct** under different top model component masking strategies. Our approach shows lower Top-k disruption and higher Top-k recovery, indicating more accurate identification of important components.

Methods	Known 1000			SQuAD v2.0		
	dis. ↓	rec. ↑	total ↑	dis. ↓	rec. ↑	total ↑
Reference						
Random	19.67	26.38	6.71	34.63	34.68	0.05
Attention-based						
Last layer	7.80†	30.41	22.61	9.79	62.84	53.05
Mean	8.61	32.27	23.66	10.78	65.99†	55.21†
Rollout	19.79	20.46	0.67	27.94	56.88	28.94
Gradient-based						
Gradient	17.31	24.66	7.35	31.19	27.56	-3.63
Input×Gradient	16.31	28.18	11.87	25.65	30.44	4.79
Integrated Gradients	9.72	38.14†	28.42†	8.67	55.98	47.31
Decomposition-based						
DePass	9.64	37.53	27.89	**7.92**	53.93	46.01
Contextual mixing-based						
IFR	21.97	16.43	-5.54	32.56	52.78	20.22
Proposed Method						
Dual Path Attribution (DPA)	**7.16**	**43.23**	**36.07**	8.06†	**70.97**	**62.91**

Table 12. Faithfulness evaluation of crucial token attribution on **Mistral-7B-Instruct-v0.3** across **Known 1000** (factual knowledge), and **SQuAD v2.0** (reading comprehension). Lower disruption (*dis.*) and higher recovery (*rec.*) indicate more faithful token localization; *total* denotes their aggregate AUC score. **Bold** indicates the best result, and † the second best. Dual Path Attribution consistently achieves the highest faithfulness across datasets.

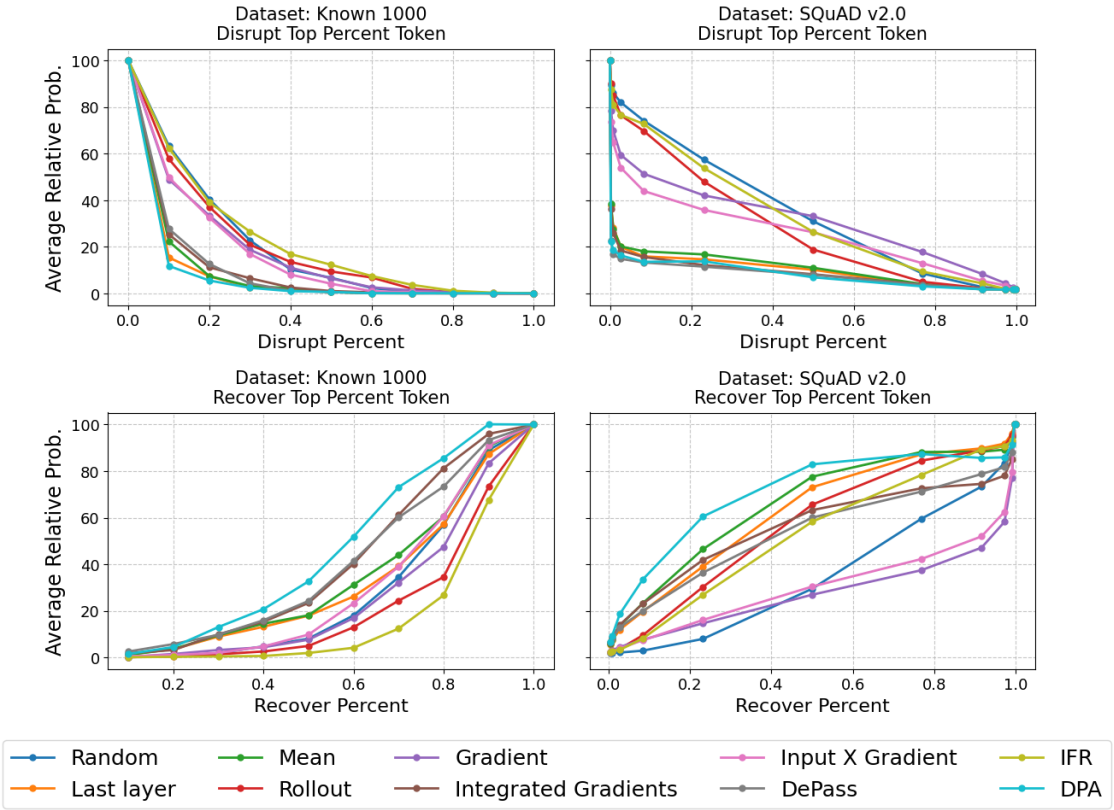

Figure 12. Performance comparison with baseline attribution methods on **Mistral-7B-Instruct-v0.3** under different top input token masking strategies. Our approach shows *lower* Top-k disruption and *higher* Top-k recovery, indicating more accurate identification of important components.

Methods	Known 1000			IOI		
	dis. ↓	rec. ↑	total ↑	dis. ↓	rec. ↑	total ↑
Reference						
Random	23.39	27.84	4.45	25.27	27.60	2.33
Baselines						
Attn-only	1.76	92.60	90.84	0.30	94.67	94.37
MLP-only	0.94	28.44	27.50	2.82	27.66	24.84
Norm-only	0.26	57.87	57.61	0.10	54.59	54.49
Gradient-based						
Gradient	0.64	29.94	29.30	0.02†	56.00	55.98
AtP	0.01†	107.73†	107.72†	**0.00**	104.50†	104.50†
Contextual mixing-based						
IFR	0.91	32.31	31.40	0.13	29.81	29.68
Proposed Method						
Dual Path Attribution (DPA)	**0.00**	**143.76**	**143.76**	**0.00**	**172.15**	**172.15**

Table 13. Component-level attribution performance on **Mistral-7B-Instruct-v0.3** across **Known 1000** (factual knowledge) and **IOI** (Indirect Object Identification) benchmarks. Lower disruption (*dis.*) and higher recovery (*rec.*) indicate more faithful localization of causally relevant components; the *total* score aggregates both effects. **Bold** denotes the best result, and † indicates the second best. Dual Path Attribution consistently achieves the most faithful component localization across benchmarks.

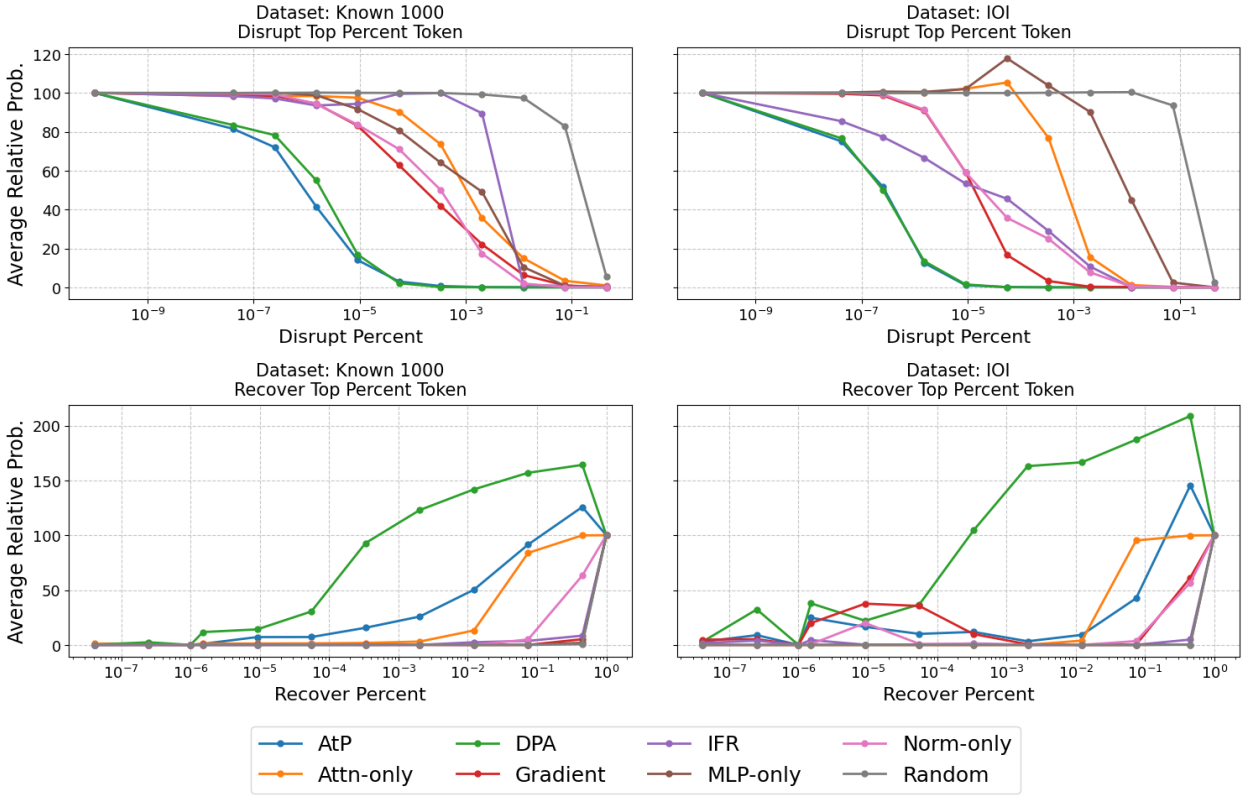

Figure 13. Performance comparison with baseline attribution methods on **Mistral-7B-Instruct-v0.3** under different top model component masking strategies. Our approach shows lower Top-k disruption and higher Top-k recovery, indicating more accurate identification of important components.