Simplifying Outcomes of Language Model Component Analyses with ELIA

Aaron Louis Eidt[1,2] **Nils Feldhus**[1,3]
[1]Technische Universität Berlin
[2]Fraunhofer Heinrich Hertz Institute
[3]BIFOLD – Berlin Institute for the Foundations of Learning and Data
`aaron.eidt@hhi.fraunhofer.de`, `feldhus@tu-berlin.de`

Abstract

While mechanistic interpretability has developed powerful tools to analyze the internal workings of Large Language Models (LLMs), their complexity has created an accessibility gap, limiting their use to specialists. We address this challenge by designing, building, and evaluating ELIA (Explainable Language Interpretability Analysis), an interactive web application that simplifies the outcomes of various language model component analyses for a broader audience. The system integrates three key techniques – Attribution Analysis, Function Vector Analysis, and Circuit Tracing – and introduces a novel methodology: using a vision-language model to automatically generate natural language explanations (NLEs) for the complex visualizations produced by these methods. The effectiveness of this approach was empirically validated through a mixed-methods user study, which revealed a clear preference for interactive, explorable interfaces over simpler, static visualizations. A key finding was that the AI-powered explanations helped bridge the knowledge gap for non-experts; a statistical analysis showed no significant correlation between a user's prior LLM experience and their comprehension scores, suggesting that the system reduced barriers to comprehension across experience levels. We conclude that an AI system can indeed simplify complex model analyses, but its true power is unlocked when paired with thoughtful, user-centered design that prioritizes interactivity, specificity, and narrative guidance.

1 Introduction

The growing capabilities of LLMs are coupled with a proportional increase in their inscrutability. While the field of mechanistic interpretability has made major strides in developing tools to reverse-engineer the internal algorithms of these black-box systems (Bereska and Gavves, 2024; Ferrando et al., 2024), a new challenge has emerged: the outputs of these analyses are often as complex as the models they seek to explain. Techniques such as attribution analysis, which traces predictions to input tokens (Sarti et al., 2023), or circuit tracing (Lindsey et al., 2025), which maps specific computational pathways, produce visualizations and data that require specialized expertise to decipher. This creates an accessibility gap, limiting the vital conversation around AI safety and reliability (Weidinger et al., 2023) to a small circle of specialists and excluding developers, domain experts, and policymakers who could benefit most.

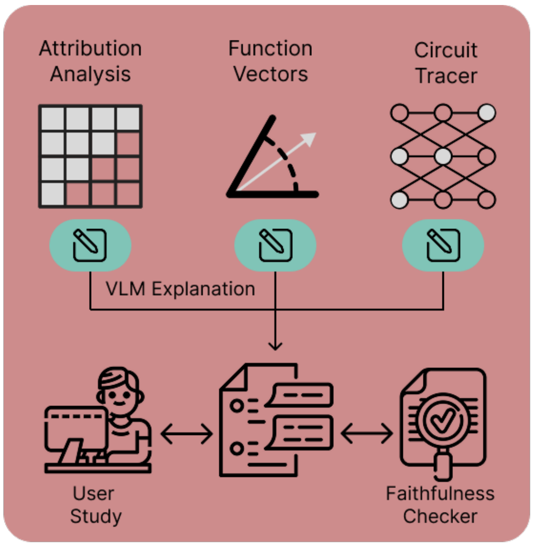


Figure 1: ELIA system overview, including three analysis methods (Attribution Analysis, Function Vectors, and Circuit Tracer) and the explanation generation workflow using VLMs to transform complex interpretability analyses into accessible NLEs. The system is evaluated using a Faithfulness Checker and a user study.

To bridge this gap, we introduce ELIA (Explainable Language Interpretability Analysis), an interactive web application[1] designed to make the outcomes of complex model analyses accessible to a broader audience. ELIA integrates three

[1]Demo: `https://hf.co/spaces/aaron0eidt/ELIA`
GitHub: `https://github.com/aaron0eidt/ELIA`

powerful interpretability techniques, Attribution Analysis (Sarti et al., 2023), Function Vector Analysis (Todd et al., 2024), and Circuit Tracing (Lindsey et al., 2025), within a user-centered interface. A vision-language model then generates NLEs for the intricate visualizations produced by these analyses.

Through a mixed-methods user study, we demonstrate the effectiveness of this approach. Our findings show that the AI-generated explanations helped reduce the knowledge gap, enabling non-experts to comprehend complex model behaviors at levels approaching those of users with prior LLM experience. Furthermore, the study revealed a strong user preference for interactive, explorable interfaces over static visualizations. This work provides empirical evidence that the strategic combination of AI-powered explanation and thoughtful, interactive design can significantly lower the barrier to understanding the internal workings of LLMs.

2 Background and Related Work

The field of **NLP interpretability** has progressed through three interconnected streams: moving from correlational to causal analysis, shifting focus from input-output attribution to internal component analysis, and developing methods to communicate these complex findings to a broader audience (Saphra and Wiegreffe, 2024; Calderon and Reichart, 2025).

Early interpretability work adapted **attribution techniques** from computer vision, such as Integrated Gradients (Sundararajan et al., 2017), to create saliency heatmaps that identify influential input tokens. However, the "attention is not explanation" debate and critical sanity checks (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Adebayo et al., 2018) revealed the limitations of these correlational methods, pushing the field toward more rigorous, intervention-based approaches.

Techniques like activation patching and causal tracing now allow researchers to establish causal links between specific model components and their behavior by **intervening in the computational process** (Zhang and Nanda, 2024). Landmark findings include the identification of induction heads that perform in-context learning (Nanda et al., 2022) and the discovery that entire tasks can be represented by abstract Function Vectors within the model's activation space (Todd et al., 2024). These vectors can be extracted and even composed, demonstrating that models learn structured, portable representations of functions.

Despite these powerful analytical tools, **communicating the findings remains a significant bottleneck**. The raw outputs, complex graphs, heatmaps, and high-dimensional plots, are often inscrutable to non-experts (Colin et al., 2022; Schuff et al., 2022). To address this, interactive visualization tools like LIT, BertViz, Inseq, and LM Transparency Tool provide explorable interfaces for experts (Tenney et al., 2020; Vig, 2019; Sarti et al., 2023; Tufanov et al., 2024). More recently, the focus has shifted to **automated explanation** systems that use explainer models to generate natural language descriptions for neuron activity or attention patterns (Bills et al., 2023; Feldhus and Kopf, 2025), agents using vision-language models for end-to-end interpretability experiment design (Shaham et al., 2024; Kim et al., 2025), and discovering circuits that represent a particular higher-level function of a model (Wang et al., 2023; Hanna et al., 2025). However, this automation introduces a critical trade-off between the faithfulness of an explanation (how accurately it reflects the model's process) and its simplicity (Feldhus et al., 2023; Parcalabescu and Frank, 2024). Our work is situated at this frontier, aiming to bridge the gap between complex, faithful analyses and simple, accessible explanations through a combination of interactive visualization and AI-generated narrative.

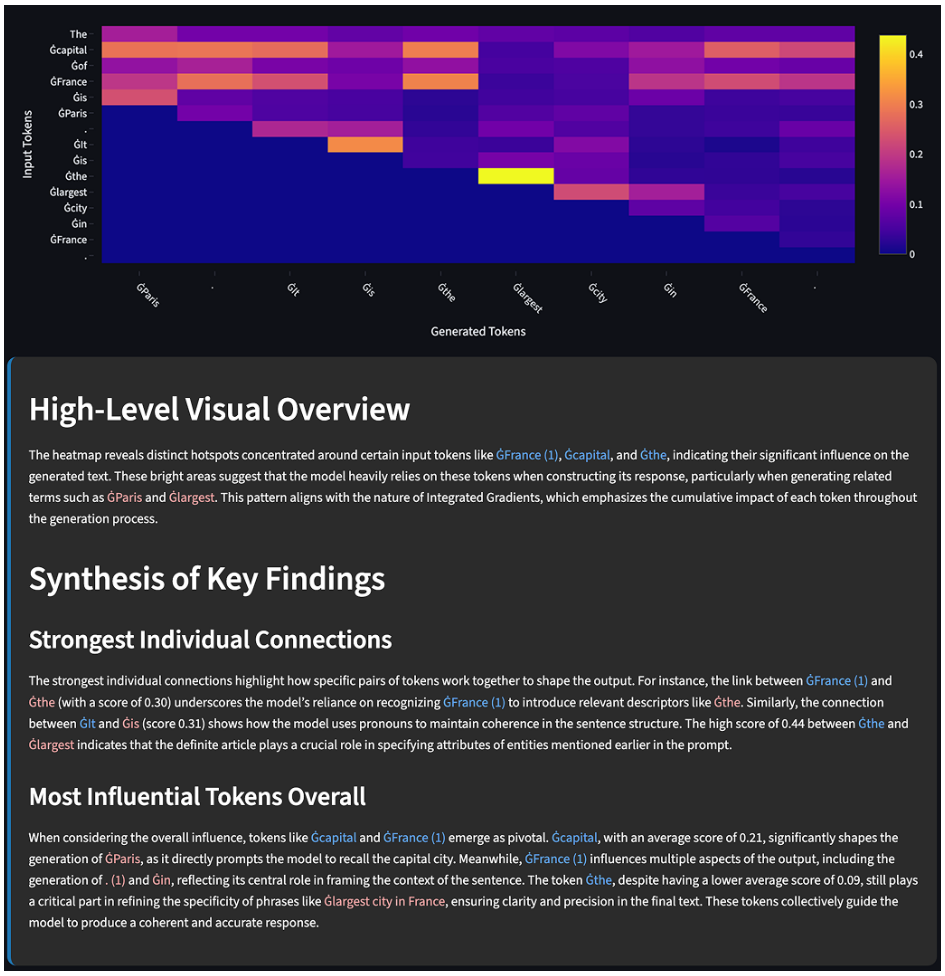
3 ELIA

3.1 System Architecture

ELIA is an interactive web application designed to make the internal mechanisms of LLMs more transparent and understandable. The system is built using Streamlit[2], a Python-based framework chosen for its ability to rapidly create data-centric, interactive user interfaces. The backend leverages the scientific Python ecosystem, with PyTorch and the Transformers library for model handling, Plotly[3] for dynamic visualizations, and the *inseq* toolkit[4] for attribution analyses (Sarti et al., 2023).

The architecture is centered around two core models: a **subject model**, the 7-billion parameter OLMo-2, whose behavior is being analyzed (Groeneveld et al., 2024); and a vision-enabled **explanation model** (Qwen2.5-VL-72B) tasked with simplifying the analytical outputs. When a user interacts with one of ELIA's three analysis pages

[2]https://streamlit.io
[3]https://plotly.com
[4]https://inseq.org



High-Level Visual Overview

The heatmap reveals distinct hotspots concentrated around certain input tokens like ĠFrance (1), Ġcapital, and Ġthe, indicating their significant influence on the generated text. These bright areas suggest that the model heavily relies on these tokens when constructing its response, particularly when generating related terms such as ĠParis and Ġlargest. This pattern aligns with the nature of Integrated Gradients, which emphasizes the cumulative impact of each token throughout the generation process.

Synthesis of Key Findings

Strongest Individual Connections

The strongest individual connections highlight how specific pairs of tokens work together to shape the output. For instance, the link between ĠFrance (1) and Ġthe (with a score of 0.30) underscores the model's reliance on recognizing ĠFrance (1) to introduce relevant descriptors like Ġthe. Similarly, the connection between ĠIt and Ġis (score 0.31) shows how the model uses pronouns to maintain coherence in the sentence structure. The high score of 0.44 between Ġthe and Ġlargest indicates that the definite article plays a crucial role in specifying attributes of entities mentioned earlier in the prompt.

Most Influential Tokens Overall

When considering the overall influence, tokens like Ġcapital and ĠFrance (1) emerge as pivotal. Ġcapital, with an average score of 0.21, significantly shapes the generation of ĠParis, as it directly prompts the model to recall the capital city. Meanwhile, ĠFrance (1) influences multiple aspects of the output, including the generation of . (1) and Ġin, reflecting its central role in framing the context of the sentence. The token Ġthe, despite having a lower average score of 0.09, still plays a critical part in refining the specificity of phrases like Ġlargest city in France, ensuring clarity and precision in the final text. These tokens collectively guide the model to produce a coherent and accurate response.

Figure 2: The interactive Attribution Heatmap using Integrated Gradients with an AI-generated natural language explanation. The heatmap visualizes the influence of input tokens on the generated output, and the explanation interprets these results in an accessible narrative.

(Attribution Analysis, Function Vector Analysis, or Circuit Tracing), the subject model's internal activations and outputs are visualized (Figure 1). These visualizations, along with structured textual data, are passed to the explanation model, following prior work on verbalizing explanations (Feldhus et al., 2023). This generates a structured, natural language summary of the key insights in an accessible narrative (Figure 2).

To ensure consistency, API calls to the explanation model are made with a low temperature and a fixed seed, making the generated text largely deterministic. The entire application is internationalized, with full support for both English and German to broaden its accessibility.

3.2 Faithfulness Verification

A key component of ELIA's architecture is an automated faithfulness verification system, designed to ensure the reliability of the AI-generated explanations. This system leverages the same explanation model in a multi-step process. First, after generating the initial narrative, the explanation model is prompted again, this time to act as a claim extraction agent, parsing its own text to identify all verifiable, factual statements and structure them as a JSON list, following a similar approach to atomic fact extraction in FActScore (Min et al., 2023). These claims range from specific quantitative statements (e.g., "Layer 12 had the highest activation.") to more abstract semantic assertions (e.g., "Early layers handle syntax."). In the second stage, a verification module programmatically checks each claim against the ground-truth data from the underlying analysis. For quantitative claims, this is a direct data comparison. For more abstract semantic claims, the explanation model is called a third time, now tasked to act as a fact-checker to assess the logical plausibility of the claim against the data. The outcome, a *verified* or *contradicted* status for each claim and the supporting evidence, is then presented to the user.

To mitigate the circularity risk of using Qwen2.5-VL-72B for both generation and verification, the verification module operates deterministically (temperature 0.0, fixed seed) and relies heavily on programmatic grounding rather than purely LLM-based judgments. When LLM-based semantic verification is required, the explainer model is constrained by hard-coded rules, negative constraints, and exact synonym-mapping directives, effectively preventing the model from self-affirming its own hallucinations.

3.3 Attribution Analysis

The Attribution Analysis page provides a granular view of the model's decision-making by quantifying the influence of individual input tokens on the generated output. It integrates two key features: core attribution methods and an influence tracer.

The primary analysis is grounded in three established **feature attribution** techniques, Saliency, Integrated Gradients, and Occlusion, which are implemented using the *Inseq* toolkit (Sarti et al., 2023). After the subject model generates text from a user's prompt, the chosen method computes an attribution matrix that is visualized as an interactive heatmap. To translate this complex data into an accessible narrative, the explanation model is given a multi-modal prompt. This prompt combines the heatmap image with a rule-based textual summary that highlights key data points, such as the most influential input tokens and the most affected output tokens, guiding the model to generate a comprehensive, structured explanation of the token-level interactions (Figure 2). The faithfulness of these explanations is also verified (Figure 6 in Appendix).

To further contextualize the model's output, the page includes an **Influence Tracing** feature that identifies similar documents from the model's training data by performing a k-nearest neighbors search against a pre-computed Faiss index of the Dolma dataset, the training corpus for OLMo (Douze et al., 2024). The user's prompt is embedded into the same vector space as the training documents, and the most similar examples are retrieved. This allows users to perform a form of data archaeology, exploring potential sources that may have influenced the model's response (Figure 7).

3.4 Function Vector Analysis

The Function Vector Analysis page offers a high-level, semantic view of the model's behavior. It allows users to explore how the model represents different instructions by comparing a user's prompt to a pre-computed, high-dimensional space of Function Vectors (Todd et al., 2024).

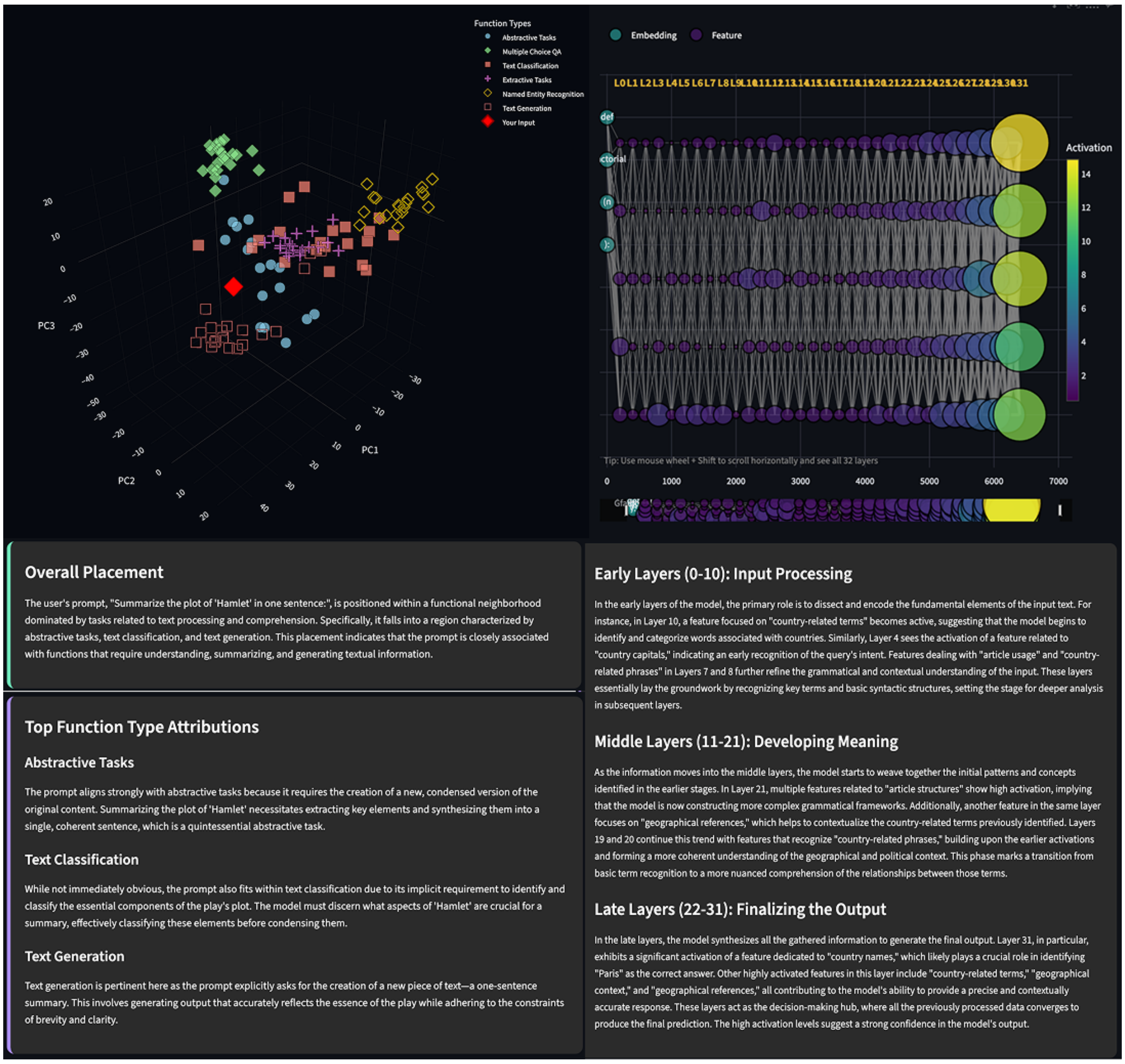
The core of this analysis is a custom dataset of instructional prompts (see Appendix B for details), organized into a hierarchy of broad "function types" (e.g., abstractive tasks) and specific "function categories" (e.g., summarization). The function vector for each category is pre-computed by averaging the final-layer, final-token activations of all its example prompts. When a user enters a new prompt, its own activation vector is computed and compared against this space using cosine similarity.

The results are presented through a suite of interactive visualizations (Figure 3, left). A 3D scatter plot, generated using Principal Component Analysis (PCA), shows the geometric relationship between the user's prompt and the function vector clusters, providing an intuitive map of the model's functional space. This is complemented by a bar chart of the top-scoring function types and a hierarchical sunburst chart that visualizes the similarity scores for all categories. For each of these visualizations, a targeted, AI-powered explanation is generated, synthesizing the key quantitative findings into an accessible, natural-language summary. The faithfulness of the PCA explanation is also verified (see Figure 8 in the Appendix).

3.5 Circuit Trace Analysis

This page offers the most granular view of the model's internal workings, building on the circuit tracing framework by (Lindsey et al., 2025). The analysis is centered around a small autoencoder, called a Cross-Layer Transcoder (CLT) (Dunefsky et al., 2024), which is pre-trained to learn a simplified, sparse representation of the OLMo model's internal activations. This CLT is trained on a diverse corpus from the Dolma dataset using L_1 sparsity regularization, gradient clipping, and cosine annealing learning rate scheduling (Figure 9 for training dynamics). The CLT is trained to reconstruct the main model's signals while being penalized for using too many features, forcing it to identify the most functionally significant patterns.

These learned features are then given semantic meaning through an automated interpretation step. For each feature, the top-activating input tokens are passed to the explanation model, which generates a concise functional label (e.g., "identifying JSON syntax"). These interpretable features become the nodes in the main visualization: a layer-



Overall Placement

The user's prompt, "Summarize the plot of 'Hamlet' in one sentence:", is positioned within a functional neighborhood dominated by tasks related to text processing and comprehension. Specifically, it falls into a region characterized by abstractive tasks, text classification, and text generation. This placement indicates that the prompt is closely associated with functions that require understanding, summarizing, and generating textual information.

Top Function Type Attributions

Abstractive Tasks

The prompt aligns strongly with abstractive tasks because it requires the creation of a new, condensed version of the original content. Summarizing the plot of 'Hamlet' necessitates extracting key elements and synthesizing them into a single, coherent sentence, which is a quintessential abstractive task.

Text Classification

While not immediately obvious, the prompt also fits within text classification due to its implicit requirement to identify and classify the essential components of the play's plot. The model must discern what aspects of 'Hamlet' are crucial for a summary, effectively classifying these elements before condensing them.

Text Generation

Text generation is pertinent here as the prompt explicitly asks for the creation of a new piece of text—a one-sentence summary. This involves generating output that accurately reflects the essence of the play while adhering to the constraints of brevity and clarity.

Early Layers (0-10): Input Processing

In the early layers of the model, the primary role is to dissect and encode the fundamental elements of the input text. For instance, in Layer 10, a feature focused on "country-related terms" becomes active, suggesting that the model begins to identify and categorize words associated with countries. Similarly, Layer 4 sees the activation of a feature related to "country capitals," indicating an early recognition of the query's intent. Features dealing with "article usage" and "country-related phrases" in Layers 7 and 8 further refine the grammatical and contextual understanding of the input. These layers essentially lay the groundwork by recognizing key terms and basic syntactic structures, setting the stage for deeper analysis in subsequent layers.

Middle Layers (11-21): Developing Meaning

As the information moves into the middle layers, the model starts to weave together the initial patterns and concepts identified in the earlier stages. In Layer 21, multiple features related to "article structures" show high activation, implying that the model is now constructing more complex grammatical frameworks. Additionally, another feature in the same layer focuses on "geographical references," which helps to contextualize the country-related terms previously identified. Layers 19 and 20 continue this trend with features that recognize "country-related phrases," building upon the earlier activations and forming a more coherent understanding of the geographical and political context. This phase marks a transition from basic term recognition to a more nuanced comprehension of the relationships between those terms.

Late Layers (22-31): Finalizing the Output

In the late layers, the model synthesizes all the gathered information to generate the final output. Layer 31, in particular, exhibits a significant activation of a feature dedicated to "country names," which likely plays a crucial role in identifying "Paris" as the correct answer. Other highly activated features in this layer include "country-related terms," "geographical context," and "geographical references," all contributing to the model's ability to provide a precise and contextually accurate response. These layers act as the decision-making hub, where all the previously processed data converges to produce the final prediction. The high activation levels suggest a strong confidence in the model's output.

Figure 3: Function Vector and Circuit Trace Analysis visualizations. The 3D PCA plot (left) places the user's prompt in a semantic functional space, while the Circuit Graph (right) traces the flow of information through interpretable features across layers. Both are accompanied by AI-generated explanations.

by-layer Circuit Graph (Figure 3, right). This graph shows the flow of information from the input tokens, through the activated features, to the final output, with node size and color indicating activation strength and edge thickness representing influence. Additionally, the system provides a view of local path ablations (Figure 10 in the Appendix), which demonstrates what happens when specific paths in the top feature graph are ablated. To make this complex graph accessible, a multi-modal prompt containing the graph image and a summary of key feature activity is used to generate a structured, AI-powered narrative of the information flow. The page also includes interactive "Subnetwork" and "Feature" explorers (see Figures 12 and 13 in the Appendix), allowing users to drill down into the behavior of individual features and their local computational pathways, each augmented with its own targeted, AI-generated explanation. The faithfulness of these explanations is verified and displayed to users (Figure 11 in the Appendix).

4 Faithfulness Analysis

Maintaining fidelity between the automatically generated narratives and the underlying model behavior requires dedicated instrumentation on every analysis page. We implement specific verification pipelines that inspect the data powering each visualization, run deterministic checks, and produce aggregate diagnostics that we summarize in Table 1 and describe in the following for every explanation.

Component	Feature	Verified	Faith. (%)
Attribution	Saliency	59/68	86.8%
	Int. Gradients	56/61	91.8%
	Occlusion	66/75	88.0%
Func. Vectors	Placement	23/24	95.8%
	Func. Type	47/47	100.0%
	Categories	44/48	91.7%
	Layer Evol.	36/36	100.0%
Circuits	Overview	45/46	97.8%
	Subnetwork	132/137	96.4%
	Features	120/125	96.0%

Table 1: Faithfulness verification results of each explanation type across all three analysis pages. All explanations are generated by Qwen2.5-VL-72B and verified against ground-truth data from the underlying analyses.

Attribution Analysis Each time a user runs any attribution method, we collect the raw attribution matrix, compute per-token peak and mean contributions, and identify the strongest interactions between input and output tokens. These statistics drive the automatically generated explanation while also feeding the Faithfulness Checker.

Function Vector Analysis The Function Vector workflow exports three independent checkpoints. First, the similarity rankings for function types and categories are recomputed from the cached activations, ensuring that any statement about top matches reflects the actual cosine ordering. Second, the PCA narrative is compared with the cluster centroids that underpin the 3D plot, and a verifier must agree that the textual summary follows from those coordinates. Third, descriptions of layer evolution are cross-checked against the activation norms and change magnitudes from the forward pass.

Circuit Trace Analysis Circuit tracing explanations operate over graphs extracted from the Cross-Layer Transcoder. For every narrative, we therefore repeat three stages: we confirm structural statements by inspecting the underlying graph (e.g., upstream/downstream connectivity, active features), validate numeric assertions by reading the stored activation values, and run a semantic check that ensures qualitative summaries remain consistent with the graph evidence. Since the same pipeline is applied to the main circuit view, the feature explorer, and the subnetwork explorer, we obtain uniformly perfect scores for the benchmark prompts.

To further validate the causal relevance of the discovered circuits, we perform intervention experiments (Figure 4). We use a set of exemplary prompts covering knowledge retrieval, code generation, and literary analysis, and by ablating the

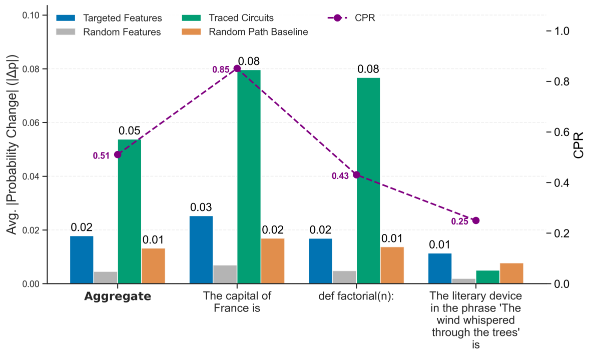


Figure 4: Impact of intervention on model output probability ($|\Delta p|$). We compare the effect of ablating top-k targeted features and traced circuits against random baselines (ablating random features or edges).

features and paths identified by the CLT, there is a substantially larger impact on the model's output probability compared to random baselines. This confirms that the system successfully isolates functionally critical components. We also compute the Circuit Performance Ratio (CPR) metric (Mueller et al., 2025), which quantifies how well a circuit recovers model performance as a function of the fraction of circuit components included.

5 Use Case: Synergistic Analysis of Knowledge Retrieval

To demonstrate how ELIA's three components can work in concert, we consider a typical knowledge retrieval prompt: "The capital of France is".

First, the *Attribution Analysis* identifies the token "France" as having the highest saliency score, indicating that the model attends to the subject.

Second, the *Function Vector Analysis* projects the prompt's activation into the pre-computed semantic space, locating it within the "Abstractive Tasks" cluster. Specifically, it scores highly on the "Next Item" and "Country Capitals" categories, suggesting that the model's internal state aligns with the high-level task type of factual completion.

Finally, the *Circuit Trace Analysis* reveals the mechanism. In early layers, features related to "article usage" and "country-related information" are active, processing the basic syntax and geographical context. In middle layers, features for "country-related terms" become prominent, linking the context to specific terminology. In late layers, the model synthesizes this information, with high activation in features related to "Geographical knowledge" and "country-related phrases". This progression shows a systematic buildup from basic grammar to sophisticated geographical knowledge,

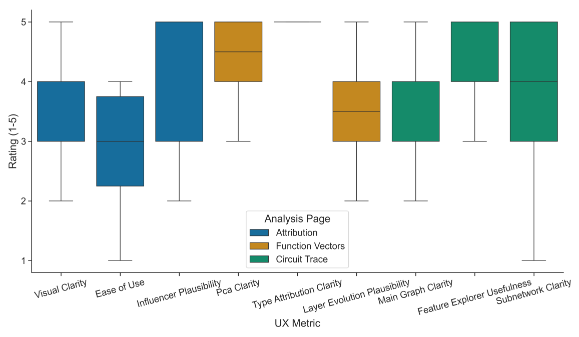


Figure 5: Grouped boxplots of UX ratings for all participants across the three analysis pages.

explaining how the model generates the answer.

This multi-layered approach allows users to build a more comprehensive understanding by combining evidence from different interpretability methods: establishing input dependence, checking semantic alignment, and inspecting components.

6 User Study

To empirically evaluate the effectiveness of the explanations and the overall usability of ELIA, a within-subjects user study was conducted with 18 undergraduate computer science students with mostly novice or intermediate experience with LLMs, and designed to assess subjective user experience (UX) and objective comprehension gains. The average completion time was approx. 1h.

6.1 Quantitative Results

Participants rated each of the three analysis pages on a 5-point Likert scale according to the following properties: visual clarity, ease of use, plausibility. As shown in Figure 5, the Function Vectors and Circuit Trace pages received significantly more positive UX ratings than the Attribution Analysis page (Kruskal-Wallis H-test, $p = 0.006$). The 'PCA Clarity' metric on the Function Vectors page received a perfect median score of 5, while the 'Ease of Use' for the Attribution Analysis heatmap received the lowest median score of the study (3), indicating it was confusing for users.

To measure objective comprehension, participants answered three multiple-choice questions per page. While there was a positive trend, a Spearman correlation test found no statistically significant relationship between a user's prior LLM experience and their correctness score ($\rho = 0.30, p = 0.23$). The average correctness scores were high across all groups: *Experts* achieved a perfect score of 1.00, *Intermediates* scored 0.98, and even *Novices* reached 0.95. This minimal performance gap suggests that the system helped reduce barriers to comprehension, enabling users with little to no prior knowledge to understand complex model behaviors at levels approaching those of more experienced users.

6.2 Qualitative Findings

Analysis of user interview transcripts revealed several key themes. A primary motivation for users was the desire to understand the black box, and the tool was most effective when it provided clear, causal explanations. However, a central challenge was the tension between detail and simplicity; users were often overwhelmed by visualizations that presented too much information at once, such as the main circuit graph.

There is also a clear user preference for interactive visualizations with strong conceptual metaphors. The 3D PCA plot and the Subnetwork Explorer were consistently praised as "very clear", while more abstract, static visuals like heatmaps were found to be confusing. Finally, a recurring theme was the need for more integrated, automated guidance. Users frequently relied on the facilitator for context and suggested that adding summaries, tooltips, and adaptive complexity would significantly improve the tool's accessibility.

7 Conclusion

ELIA shows that the tools of mechanistic interpretability can become approachable, interactive experiences without giving up analytic depth. By combining the complementary views from attribution heatmaps, function vector projections, and circuit tracing graphs with structured AI narratives and automated faithfulness checks, the platform closes a long-standing accessibility gap: participants in our study reached similar comprehension regardless of prior LLM experience and clearly preferred the explorable interfaces that ELIA provides. The quantitative gains, qualitative feedback, and high verification scores together suggest that interpretability workflows can feel like guided investigations instead of expert-only diagnostics. We hope to encourage the interpretability community to treat usability and faithfulness as co-equal concerns, nudging the field toward tools that invite participation rather than gatekeep expertise.

Limitations

ELIA is currently limited to two languages (English, German), OLMo as the explained model, and the Qwen-VL model as the explainer model. Richer intervention tools (e.g., counterfactual editing or causal scrubbing) might be necessary to provide a comprehensive user-centric view on interpreting language model behavior. The user study was limited to subjective ratings and short-term interactions, while studying longer-term usage in professional settings remains a necessary future direction to prove the advantages of ELIA we have so far recorded.

Ethical Statement

The participants in the user study were compensated at or above the minimum wage, in accordance with the standards of our host institutions' regions.

Acknowledgments

We thank the reviewers of the EACL 2026 System Demonstrations track for their valuable feedback. This research is funded by the Berlin Institute for the Foundations of Learning and Data (BIFOLD, ref. 01IS18037A)

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, volume 31, pages 9505–9515. Curran Associates, Inc.

Leonard Bereska and Stratis Gavves. 2024. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*. Survey Certification, Expert Certification.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. OpenAI technical report.

Nitay Calderon and Roi Reichart. 2025. On behalf of the stakeholders: Trends in NLP model interpretability in the era of LLMs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 656–693, Albuquerque, New Mexico. Association for Computational Linguistics.

Julien Colin, Thomas Fel, Rémi Cadène, and Thomas Serre. 2022. What I cannot predict, I do not understand: A human-centered evaluation framework for explainability methods. In *Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv Preprints*.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. 2024. Transcoders find interpretable LLM feature circuits. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Nils Feldhus, Leonhard Hennig, Maximilian Dustin Nasert, Christopher Ebert, Robert Schwarzenberg, and Sebastian Möller. 2023. Saliency map verbalization: Comparing feature importance representations from model-free and instruction-based methods. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 30–46, Toronto, Canada. Association for Computational Linguistics.

Nils Feldhus and Laura Kopf. 2025. Interpreting language models through concept descriptions: A survey. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 149–162, Suzhou, China. Association for Computational Linguistics.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *arXiv*, abs/2405.00208.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. 2025. Circuit-tracer: A new library for finding feature circuits. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 239–249, Suzhou, China. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Been Kim, John Hewitt, Neel Nanda, Noah Fiedel, and Oyvind Tafjord. 2025. Because we have llms, we can and should pursue agentic interpretability. *arXiv*, abs/2506.12152.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, and 1 others. 2025. On the biology of a large language model. *Anthropic*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12076–12100. Association for Computational Linguistics.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, and 4 others. 2025. MIB: A mechanistic interpretability benchmark. In *Forty-second International Conference on Machine Learning*.

Neel Nanda, Nelson Elhage, Catherine Olsson, and 1 others. 2022. In-context learning and induction heads. Transformer Circuits thread.

Letitia Parcalabescu and Anette Frank. 2024. On measuring faithfulness or self-consistency of natural language explanations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6048–6089, Bangkok, Thailand. Association for Computational Linguistics.

Naomi Saphra and Sarah Wiegreffe. 2024. Mechanistic? In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 480–498, Miami, Florida, US. Association for Computational Linguistics.

Gabriele Sarti, Nils Feldhus, Ludwig Sickert, and Oskar van der Wal. 2023. Inseq: An interpretability toolkit for sequence generation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 421–435, Toronto, Canada. Association for Computational Linguistics.

Hendrik Schuff, Alon Jacovi, Heike Adel, Yoav Goldberg, and Ngoc Thang Vu. 2022. Human interpretation of saliency-based explanation over text. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 611–636, New York, NY, USA. Association for Computing Machinery.

Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. 2024. A multimodal automated interpretability agent. In *Forty-first International Conference on Machine Learning*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 107–118, Online. Association for Computational Linguistics.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*.

Igor Tufanov, Karen Hambardzumyan, Javier Ferrando, and Elena Voita. 2024. Lm transparency tool: Interactive tool for analyzing transformer language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (System Demonstrations)*, pages 51–60. Association for Computational Linguistics. Open-source code available at `https://github.com/facebookresearch/llm-transparency-tool`.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

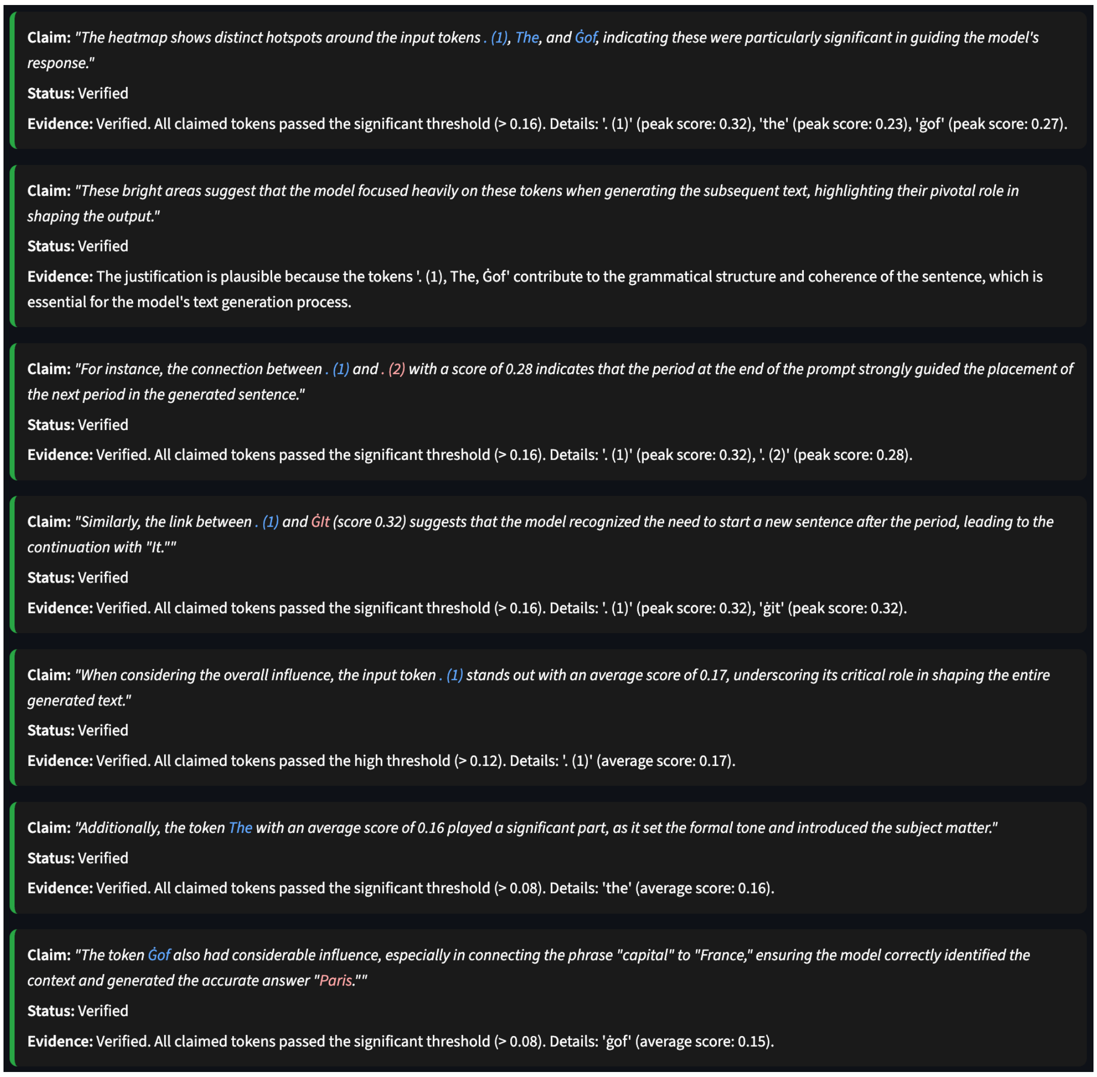
Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Laura Weidinger, Maribeth Rauh, Nahema Marchal, Arianna Manzini, Lisa Anne Hendricks, Juan Mateos-Garcia, Stevie Bergman, Jackie Kay, Conor Griffin, Ben Bariach, Iason Gabriel, Verena Rieser, and William Isaac. 2023. Sociotechnical safety evaluation of generative ai systems. *arXiv*, abs/2310.11986.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

A Attribution Analysis



Claim: *"The heatmap shows distinct hotspots around the input tokens . (1), The, and Ġof, indicating these were particularly significant in guiding the model's response."*

Status: Verified

Evidence: Verified. All claimed tokens passed the significant threshold (> 0.16). Details: '. (1)' (peak score: 0.32), 'the' (peak score: 0.23), 'ġof' (peak score: 0.27).

Claim: *"These bright areas suggest that the model focused heavily on these tokens when generating the subsequent text, highlighting their pivotal role in shaping the output."*

Status: Verified

Evidence: The justification is plausible because the tokens '. (1), The, Ġof' contribute to the grammatical structure and coherence of the sentence, which is essential for the model's text generation process.

Claim: *"For instance, the connection between . (1) and . (2) with a score of 0.28 indicates that the period at the end of the prompt strongly guided the placement of the next period in the generated sentence."*

Status: Verified

Evidence: Verified. All claimed tokens passed the significant threshold (> 0.16). Details: '. (1)' (peak score: 0.32), '. (2)' (peak score: 0.28).

Claim: *"Similarly, the link between . (1) and ĠIt (score 0.32) suggests that the model recognized the need to start a new sentence after the period, leading to the continuation with "It.""*

Status: Verified

Evidence: Verified. All claimed tokens passed the significant threshold (> 0.16). Details: '. (1)' (peak score: 0.32), 'ġit' (peak score: 0.32).

Claim: *"When considering the overall influence, the input token . (1) stands out with an average score of 0.17, underscoring its critical role in shaping the entire generated text."*

Status: Verified

Evidence: Verified. All claimed tokens passed the high threshold (> 0.12). Details: '. (1)' (average score: 0.17).

Claim: *"Additionally, the token The with an average score of 0.16 played a significant part, as it set the formal tone and introduced the subject matter."*

Status: Verified

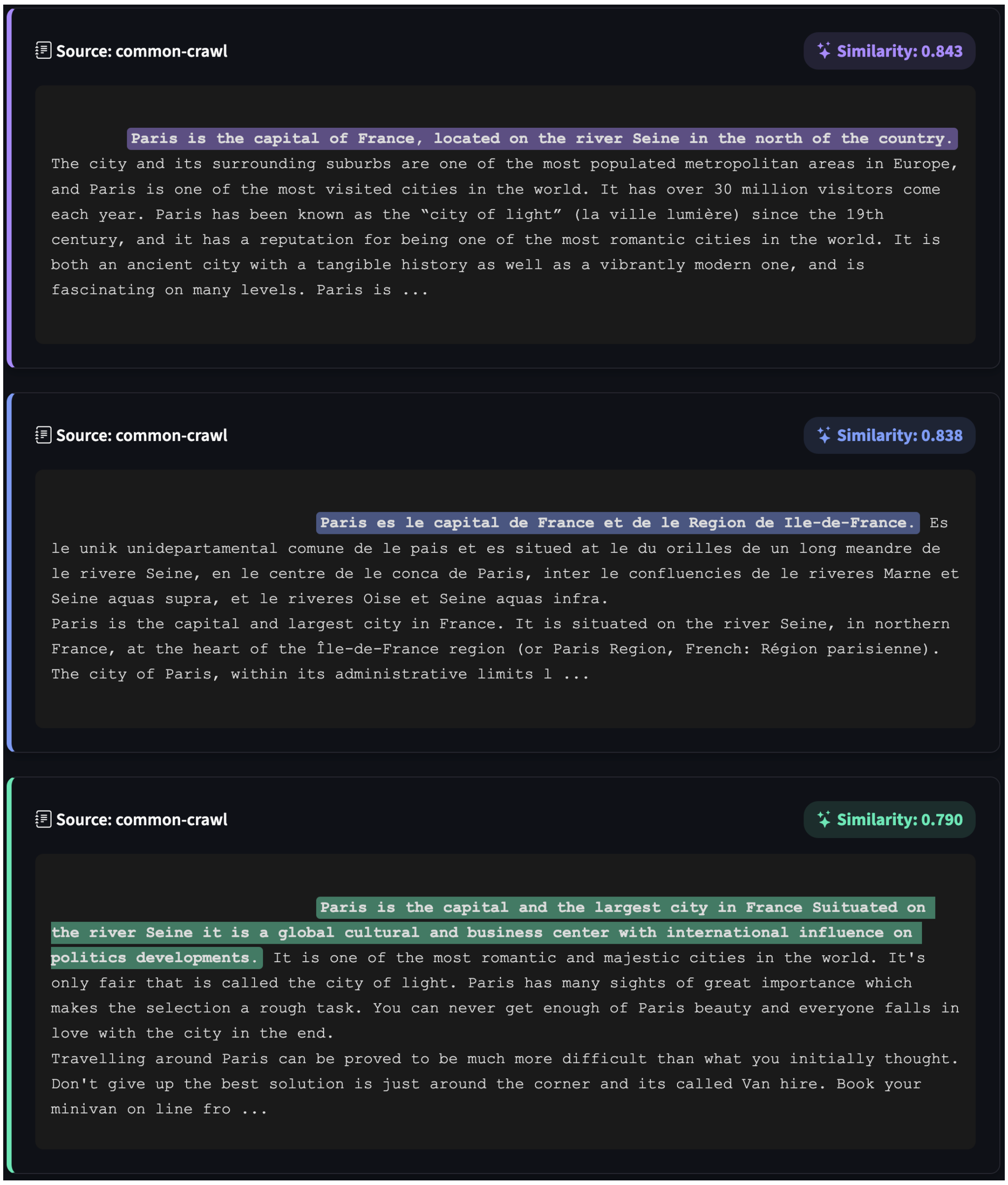
Evidence: Verified. All claimed tokens passed the significant threshold (> 0.08). Details: 'the' (average score: 0.16).

Claim: *"The token Ġof also had considerable influence, especially in connecting the phrase "capital" to "France," ensuring the model correctly identified the context and generated the accurate answer "Paris.""*

Status: Verified

Evidence: Verified. All claimed tokens passed the significant threshold (> 0.08). Details: 'ġof' (average score: 0.15).

Figure 6: Attribution Analysis: Faithfulness verification of the AI-generated explanation.



Source: common-crawl

Similarity: 0.843

```
Paris is the capital of France, located on the river Seine in the north of the country.
The city and its surrounding suburbs are one of the most populated metropolitan areas in Europe,
and Paris is one of the most visited cities in the world. It has over 30 million visitors come
each year. Paris has been known as the "city of light" (la ville lumière) since the 19th
century, and it has a reputation for being one of the most romantic cities in the world. It is
both an ancient city with a tangible history as well as a vibrantly modern one, and is
fascinating on many levels. Paris is ...
```

Source: common-crawl

Similarity: 0.838

```
Paris es le capital de France et de le Region de Ile-de-France. Es
le unik unidepartamental comune de le pais et es situed at le du orilles de un long meandre de
le rivere Seine, en le centre de le conca de Paris, inter le confluencies de le riveres Marne et
Seine aquas supra, et le riveres Oise et Seine aquas infra.
Paris is the capital and largest city in France. It is situated on the river Seine, in northern
France, at the heart of the Île-de-France region (or Paris Region, French: Région parisienne).
The city of Paris, within its administrative limits 1 ...
```

Source: common-crawl

Similarity: 0.790

```
Paris is the capital and the largest city in France Suituated on
the river Seine it is a global cultural and business center with international influence on
politics developments. It is one of the most romantic and majestic cities in the world. It's
only fair that is called the city of light. Paris has many sights of great importance which
makes the selection a rough task. You can never get enough of Paris beauty and everyone falls in
love with the city in the end.
Travelling around Paris can be proved to be much more difficult than what you initially thought.
Don't give up the best solution is just around the corner and its called Van hire. Book your
minivan on line fro ...
```

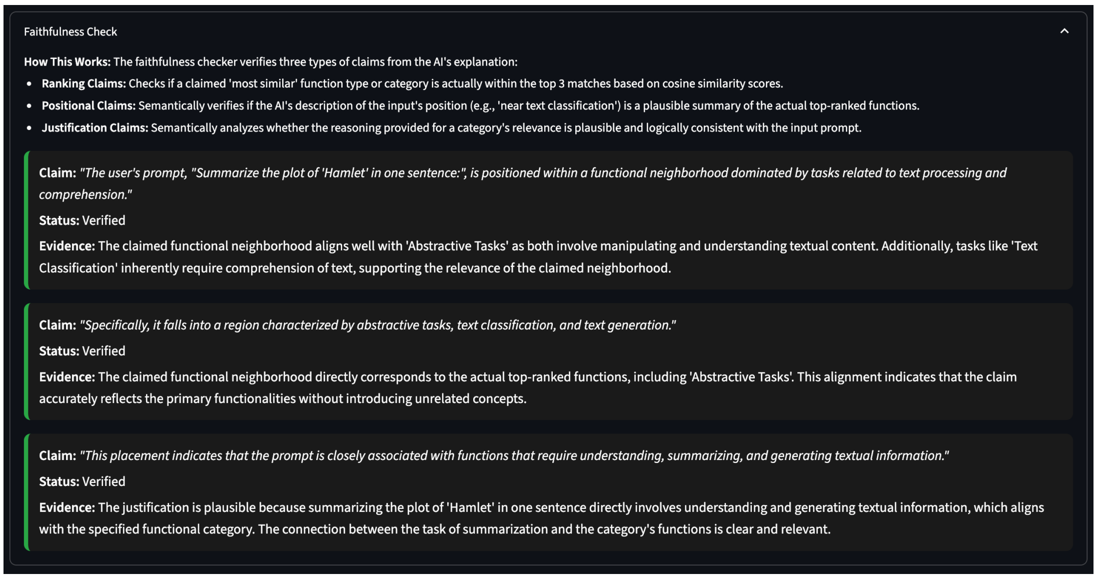
Figure 7: Influence Tracer: Retrieves and displays training documents similar to the prompt.

B Function Vector Analysis

The Function Vector space is constructed using a custom bilingual dataset (English and German) to ensure robust, cross-lingual functional mappings. For each category, we pass the instructional prompts through OLMo-2-1124-7B and extract the activation vector from the final token position of the last hidden layer. The definitive function vector for a given category is computed as the mean of these activations. To maintain interactivity and minimize computational overhead in the live application, this high-dimensional space is pre-computed offline. During real-time usage, analyzing a new prompt requires only a single forward pass to extract its activation vector, followed by a highly efficient cosine similarity comparison against the pre-cached semantic space.

Function Type	Category	Example Prompts
Abstractive Tasks	`country_capital`	The capital of France is What city serves as the capital of Japan?
	`translation_german`	Translating 'hello' into German gives What would a German speaker say for 'world'?
	`next_item`	After 'Monday' comes The number following 5 is
Multiple Choice QA	`commonsense_qa`	What happens when you mix red and blue? Why do people wear coats in winter?
	`math_qa`	What is 15 multiplied by 8? Calculate the area of a square with side 5
	`geography_qa`	Which is the largest ocean? What is the longest river in the world?
Text Classification	`sentiment_analysis`	Is this text positive or negative? What emotion does this express?
	`language_detection`	What language is this text written in? Identify the language of this sentence
	`spam_detection`	Is this message spam? Classify this email as spam or legitimate
Extractive Tasks	`adjective_vs_verb`	Is 'running' an adjective or verb? Classify 'beautiful' as adjective or verb
	`living_vs_nonliving`	Is 'tree' living or non-living? Classify 'car' as living or non-living
	`concrete_vs_abstract`	Is 'happiness' concrete or abstract? Classify 'table' as concrete or abstract
Named Entity Recognition	`ner_person`	Identify the person name in this text Extract all person names mentioned
	`ner_location`	What location is mentioned here? Extract all place names from the text
	`ner_organization`	Find the organization name Extract company or institution names
Text Generation	`complete_sentence`	Complete this sentence: "The weather today is" Finish the thought: "In the future, we will"
	`continue_story`	Continue the story: "Once upon a time..." What happens next in this story?
	`question_generation`	Generate a question about this topic Create a question based on this text

Table 2: Overview of the Function Vector dataset structure. The dataset contains 6 function types with 120 total categories. Each category includes 5 example prompts. This table shows representative examples from each function type.



Faithfulness Check

How This Works: The faithfulness checker verifies three types of claims from the AI's explanation:

- **Ranking Claims:** Checks if a claimed 'most similar' function type or category is actually within the top 3 matches based on cosine similarity scores.
- **Positional Claims:** Semantically verifies if the AI's description of the input's position (e.g., 'near text classification') is a plausible summary of the actual top-ranked functions.
- **Justification Claims:** Semantically analyzes whether the reasoning provided for a category's relevance is plausible and logically consistent with the input prompt.

Claim: *"The user's prompt, "Summarize the plot of 'Hamlet' in one sentence:", is positioned within a functional neighborhood dominated by tasks related to text processing and comprehension."*

Status: Verified

Evidence: The claimed functional neighborhood aligns well with 'Abstractive Tasks' as both involve manipulating and understanding textual content. Additionally, tasks like 'Text Classification' inherently require comprehension of text, supporting the relevance of the claimed neighborhood.

Claim: *"Specifically, it falls into a region characterized by abstractive tasks, text classification, and text generation."*

Status: Verified

Evidence: The claimed functional neighborhood directly corresponds to the actual top-ranked functions, including 'Abstractive Tasks'. This alignment indicates that the claim accurately reflects the primary functionalities without introducing unrelated concepts.

Claim: *"This placement indicates that the prompt is closely associated with functions that require understanding, summarizing, and generating textual information."*

Status: Verified

Evidence: The justification is plausible because summarizing the plot of 'Hamlet' in one sentence directly involves understanding and generating textual information, which aligns with the specified functional category. The connection between the task of summarization and the category's functions is clear and relevant.

Figure 8: Function Vector Analysis: Faithfulness verification of the AI-generated PCA explanation.

C Circuit Trace Analysis

The Cross-Layer Transcoder (CLT) is trained offline to learn a sparse, simplified representation of the model's internal activations. The model was trained on text samples from the Dolma dataset using a batch size of 16 over 1,500 training steps. Our architecture maps the hidden dimension to 512 interpretable features per layer, utilizing a JumpReLU activation (threshold = 0.0) alongside an L_1 sparsity penalty ($\lambda = 1e^{-3}$). Optimization was performed using Adam (learning rate: $3e^{-4}$) with cosine annealing and gradient clipping (max norm: 1.0) to ensure stable convergence. The training dynamics are visualized in Figure 9.

By performing this resource-intensive training offline, the live computational cost of ELIA is drastically reduced. Real-time operations are limited to standard forward passes and asynchronous API calls for the natural language explanations, ensuring the interface remains highly interactive without requiring extensive local compute resources.

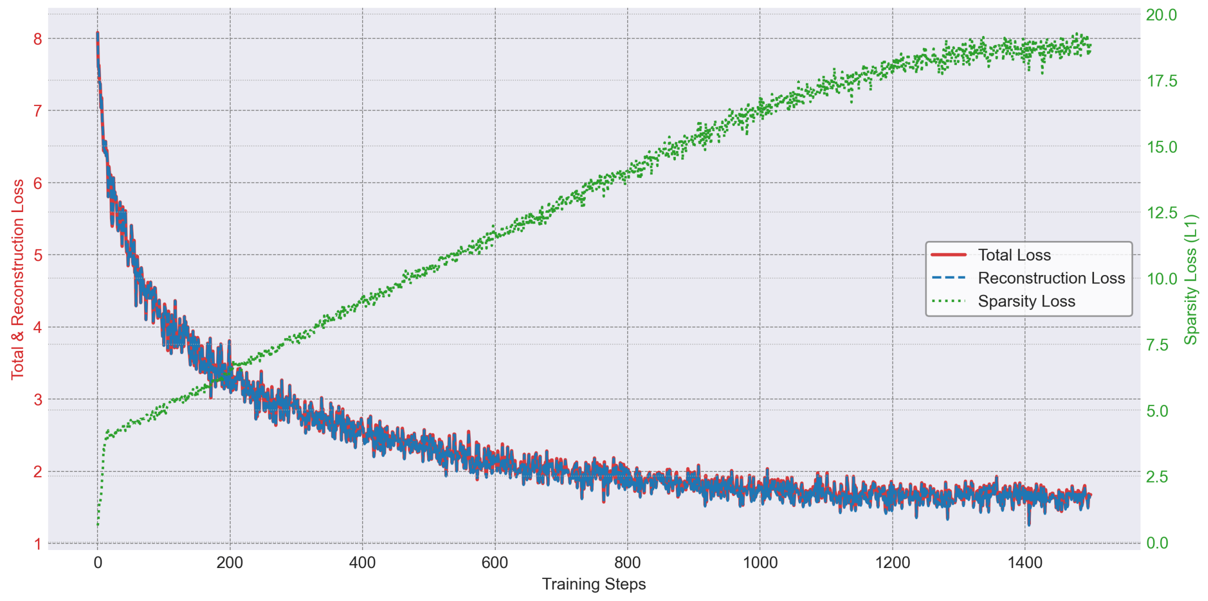
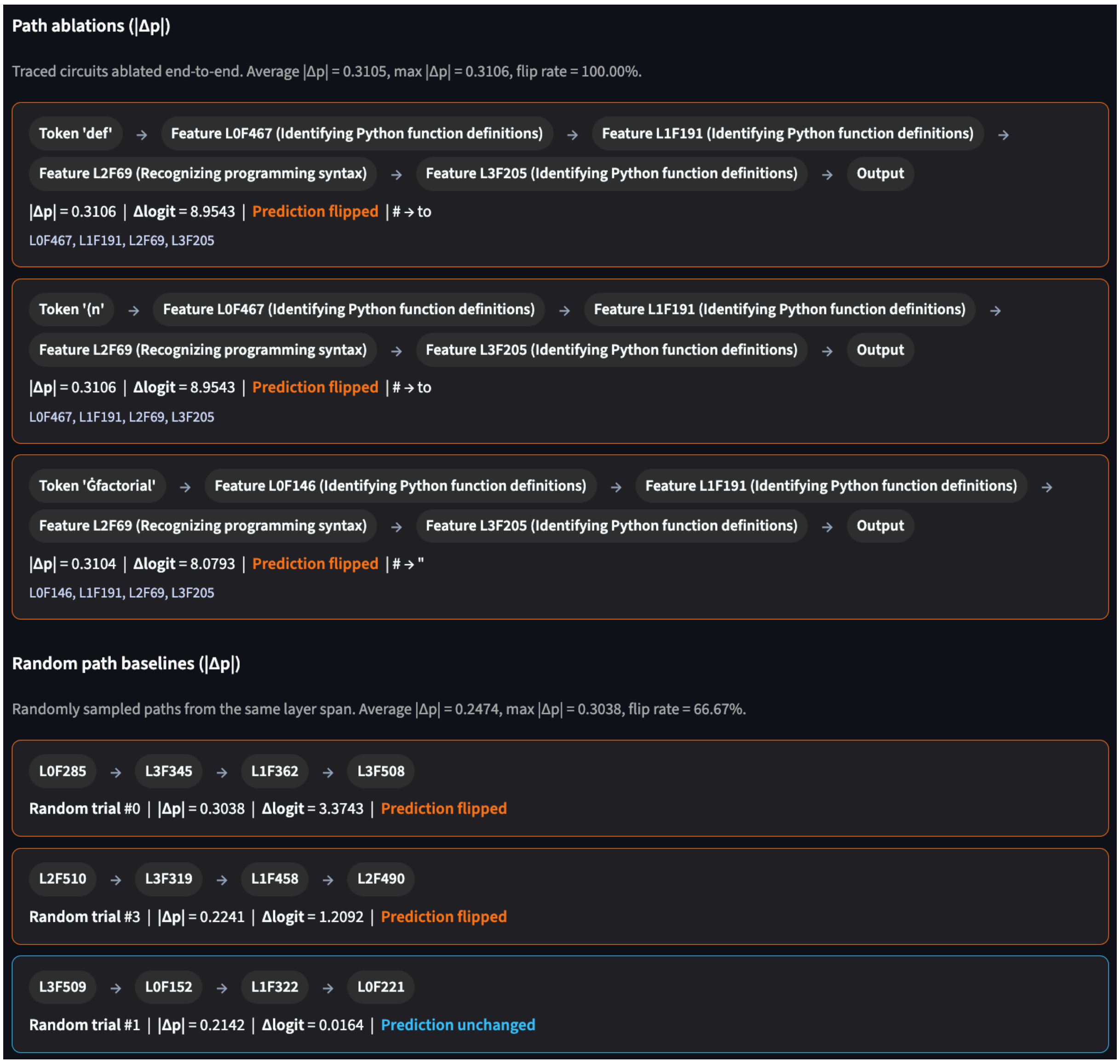


Figure 9: Cross-Layer Transcoder training dynamics showing Total Loss, Reconstruction Loss, and Sparsity Loss (L_1) over 1500 training steps. The CLT is trained on the Dolma dataset with L_1 sparsity regularization, gradient clipping, and cosine annealing learning rate scheduling.



Path ablations (|Δp|)

Traced circuits ablated end-to-end. Average |Δp| = 0.3105, max |Δp| = 0.3106, flip rate = 100.00%.

Token 'def' → Feature L0F467 (Identifying Python function definitions) → Feature L1F191 (Identifying Python function definitions) → Feature L2F69 (Recognizing programming syntax) → Feature L3F205 (Identifying Python function definitions) → Output

|Δp| = 0.3106 | Δlogit = 8.9543 | Prediction flipped | # → to

L0F467, L1F191, L2F69, L3F205

Token '(n' → Feature L0F467 (Identifying Python function definitions) → Feature L1F191 (Identifying Python function definitions) → Feature L2F69 (Recognizing programming syntax) → Feature L3F205 (Identifying Python function definitions) → Output

|Δp| = 0.3106 | Δlogit = 8.9543 | Prediction flipped | # → to

L0F467, L1F191, L2F69, L3F205

Token 'Ġfactorial' → Feature L0F146 (Identifying Python function definitions) → Feature L1F191 (Identifying Python function definitions) → Feature L2F69 (Recognizing programming syntax) → Feature L3F205 (Identifying Python function definitions) → Output

|Δp| = 0.3104 | Δlogit = 8.0793 | Prediction flipped | # → "

L0F146, L1F191, L2F69, L3F205

Random path baselines (|Δp|)

Randomly sampled paths from the same layer span. Average |Δp| = 0.2474, max |Δp| = 0.3038, flip rate = 66.67%.

L0F285 → L3F345 → L1F362 → L3F508

Random trial #0 | |Δp| = 0.3038 | Δlogit = 3.3743 | Prediction flipped

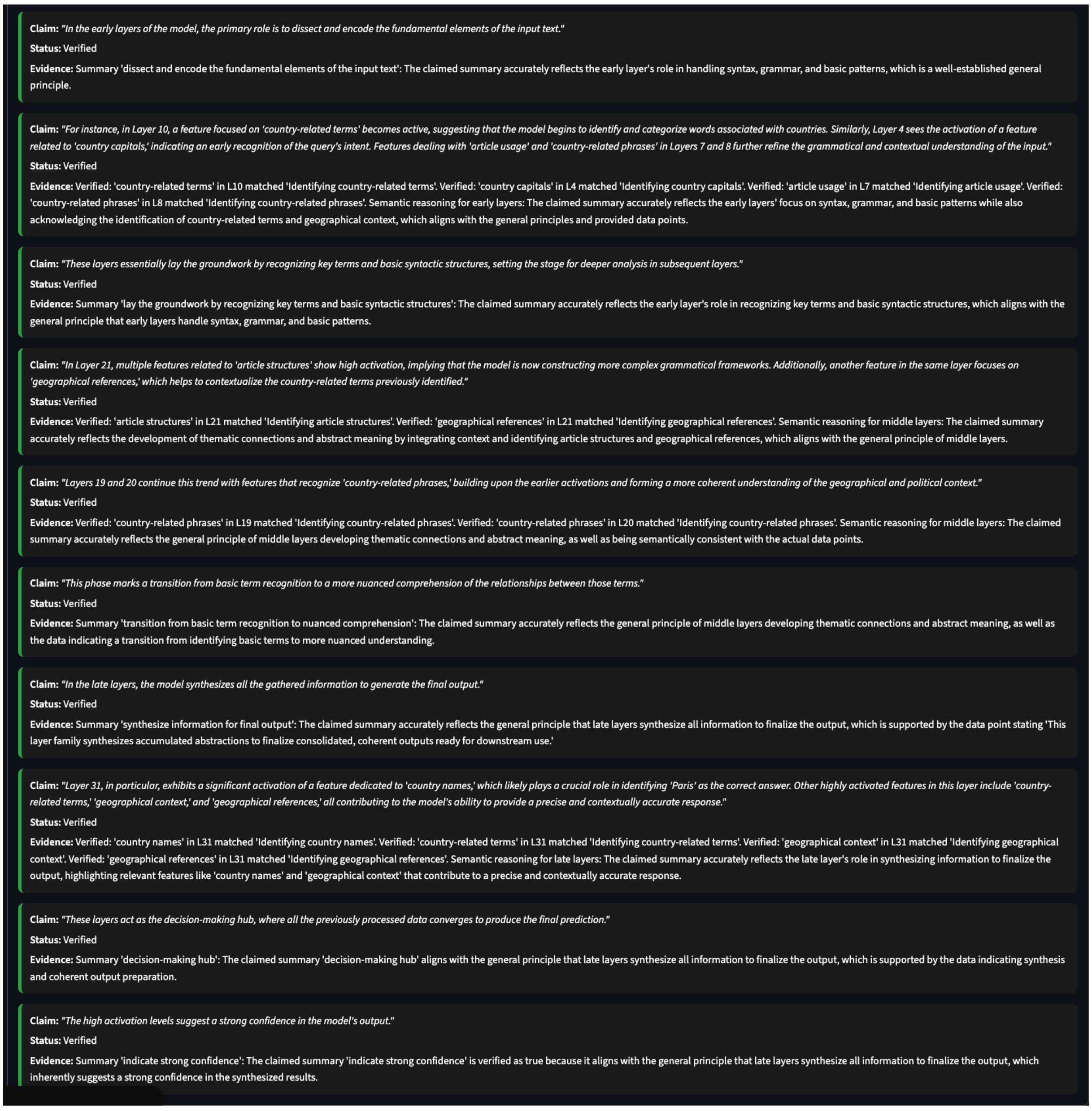
L2F510 → L3F319 → L1F458 → L2F490

Random trial #3 | |Δp| = 0.2241 | Δlogit = 1.2092 | Prediction flipped

L3F509 → L0F152 → L1F322 → L0F221

Random trial #1 | |Δp| = 0.2142 | Δlogit = 0.0164 | Prediction unchanged

Figure 10: Circuit Trace Analysis: Local path ablations. This view shows the effect of ablating specific paths within the top feature graph shown in the main circuit trace visualization.



Claim: *"In the early layers of the model, the primary role is to dissect and encode the fundamental elements of the input text."*

Status: Verified

Evidence: Summary 'dissect and encode the fundamental elements of the input text': The claimed summary accurately reflects the early layer's role in handling syntax, grammar, and basic patterns, which is a well-established general principle.

Claim: *"For instance, in Layer 10, a feature focused on 'country-related terms' becomes active, suggesting that the model begins to identify and categorize words associated with countries. Similarly, Layer 4 sees the activation of a feature related to 'country capitals,' indicating an early recognition of the query's intent. Features dealing with 'article usage' and 'country-related phrases' in Layers 7 and 8 further refine the grammatical and contextual understanding of the input."*

Status: Verified

Evidence: Verified: 'country-related terms' in L10 matched 'Identifying country-related terms'. Verified: 'country capitals' in L4 matched 'Identifying country capitals'. Verified: 'article usage' in L7 matched 'Identifying article usage'. Verified: 'country-related phrases' in L8 matched 'Identifying country-related phrases'. Semantic reasoning for early layers: The claimed summary accurately reflects the early layers' focus on syntax, grammar, and basic patterns while also acknowledging the identification of country-related terms and geographical context, which aligns with the general principles and provided data points.

Claim: *"These layers essentially lay the groundwork by recognizing key terms and basic syntactic structures, setting the stage for deeper analysis in subsequent layers."*

Status: Verified

Evidence: Summary 'lay the groundwork by recognizing key terms and basic syntactic structures': The claimed summary accurately reflects the early layer's role in recognizing key terms and basic syntactic structures, which aligns with the general principle that early layers handle syntax, grammar, and basic patterns.

Claim: *"In Layer 21, multiple features related to 'article structures' show high activation, implying that the model is now constructing more complex grammatical frameworks. Additionally, another feature in the same layer focuses on 'geographical references,' which helps to contextualize the country-related terms previously identified."*

Status: Verified

Evidence: Verified: 'article structures' in L21 matched 'Identifying article structures'. Verified: 'geographical references' in L21 matched 'Identifying geographical references'. Semantic reasoning for middle layers: The claimed summary accurately reflects the development of thematic connections and abstract meaning by integrating context and identifying article structures and geographical references, which aligns with the general principle of middle layers.

Claim: *"Layers 19 and 20 continue this trend with features that recognize 'country-related phrases,' building upon the earlier activations and forming a more coherent understanding of the geographical and political context."*

Status: Verified

Evidence: Verified: 'country-related phrases' in L19 matched 'Identifying country-related phrases'. Verified: 'country-related phrases' in L20 matched 'Identifying country-related phrases'. Semantic reasoning for middle layers: The claimed summary accurately reflects the general principle of middle layers developing thematic connections and abstract meaning, as well as being semantically consistent with the actual data points.

Claim: *"This phase marks a transition from basic term recognition to a more nuanced comprehension of the relationships between those terms."*

Status: Verified

Evidence: Summary 'transition from basic term recognition to nuanced comprehension': The claimed summary accurately reflects the general principle of middle layers developing thematic connections and abstract meaning, as well as the data indicating a transition from identifying basic terms to more nuanced understanding.

Claim: *"In the late layers, the model synthesizes all the gathered information to generate the final output."*

Status: Verified

Evidence: Summary 'synthesize information for final output': The claimed summary accurately reflects the general principle that late layers synthesize all information to finalize the output, which is supported by the data point stating 'This layer family synthesizes accumulated abstractions to finalize consolidated, coherent outputs ready for downstream use.'

Claim: *"Layer 31, in particular, exhibits a significant activation of a feature dedicated to 'country names,' which likely plays a crucial role in identifying 'Paris' as the correct answer. Other highly activated features in this layer include 'country-related terms,' 'geographical context,' and 'geographical references,' all contributing to the model's ability to provide a precise and contextually accurate response."*

Status: Verified

Evidence: Verified: 'country names' in L31 matched 'Identifying country names'. Verified: 'country-related terms' in L31 matched 'Identifying country-related terms'. Verified: 'geographical context' in L31 matched 'Identifying geographical context'. Verified: 'geographical references' in L31 matched 'Identifying geographical references'. Semantic reasoning for late layers: The claimed summary accurately reflects the late layer's role in synthesizing information to finalize the output, highlighting relevant features like 'country names' and 'geographical context' that contribute to a precise and contextually accurate response.

Claim: *"These layers act as the decision-making hub, where all the previously processed data converges to produce the final prediction."*

Status: Verified

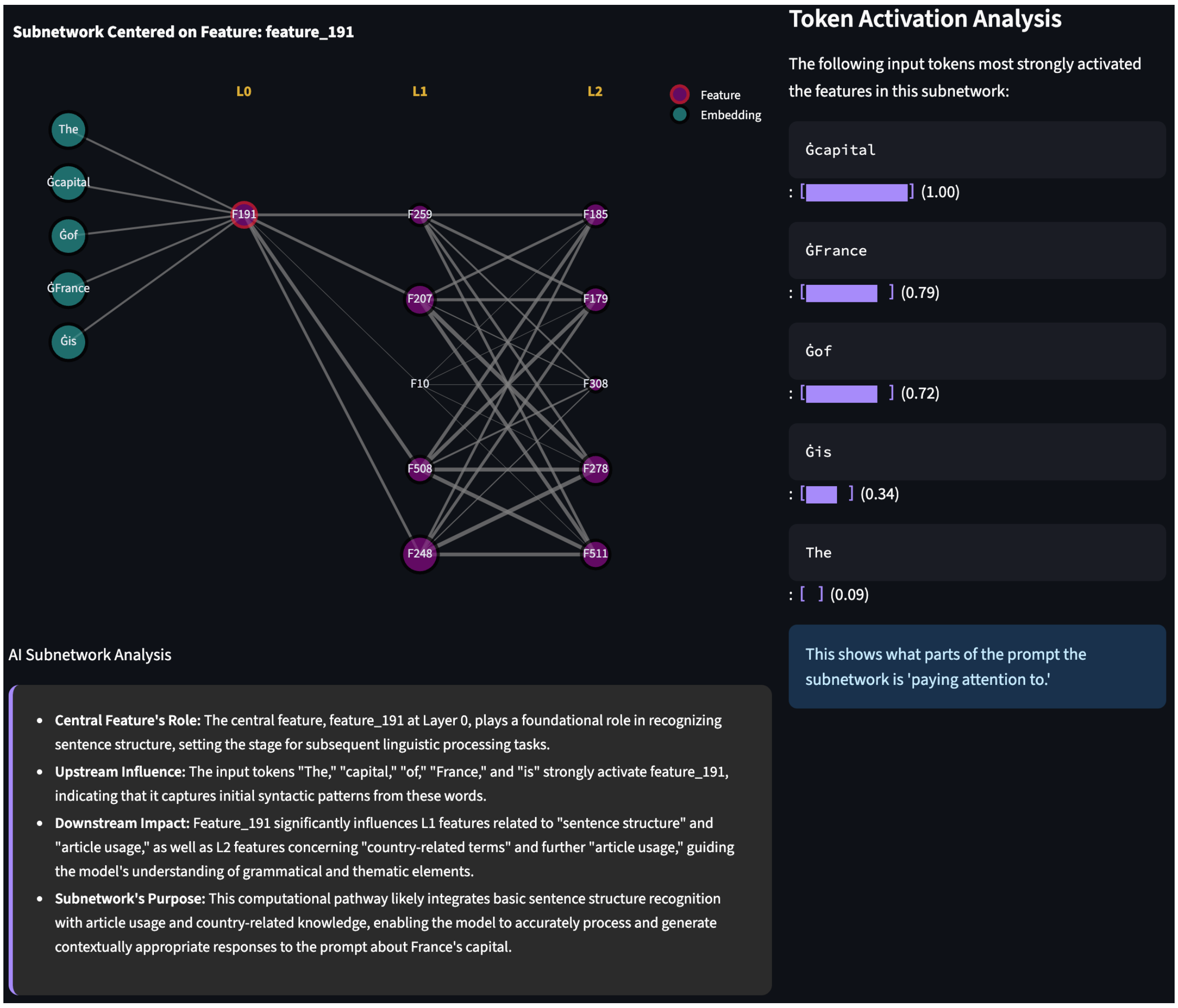
Evidence: Summary 'decision-making hub': The claimed summary 'decision-making hub' aligns with the general principle that late layers synthesize all information to finalize the output, which is supported by the data indicating synthesis and coherent output preparation.

Claim: *"The high activation levels suggest a strong confidence in the model's output."*

Status: Verified

Evidence: Summary 'indicate strong confidence': The claimed summary 'indicate strong confidence' is verified as true because it aligns with the general principle that late layers synthesize all information to finalize the output, which inherently suggests a strong confidence in the synthesized results.

Figure 11: Circuit Trace Analysis: Faithfulness verification of the circuit explanation.


Subnetwork Centered on Feature: feature_191

AI Subnetwork Analysis

- **Central Feature's Role:** The central feature, feature_191 at Layer 0, plays a foundational role in recognizing sentence structure, setting the stage for subsequent linguistic processing tasks.
- **Upstream Influence:** The input tokens "The," "capital," "of," "France," and "is" strongly activate feature_191, indicating that it captures initial syntactic patterns from these words.
- **Downstream Impact:** Feature_191 significantly influences L1 features related to "sentence structure" and "article usage," as well as L2 features concerning "country-related terms" and further "article usage," guiding the model's understanding of grammatical and thematic elements.
- **Subnetwork's Purpose:** This computational pathway likely integrates basic sentence structure recognition with article usage and country-related knowledge, enabling the model to accurately process and generate contextually appropriate responses to the prompt about France's capital.

Token Activation Analysis

The following input tokens most strongly activated the features in this subnetwork:

`Ġcapital`

: [██████████] (1.00)

`ĠFrance`

: [███████] (0.79)

`Ġof`

: [███████] (0.72)

`Ġis`

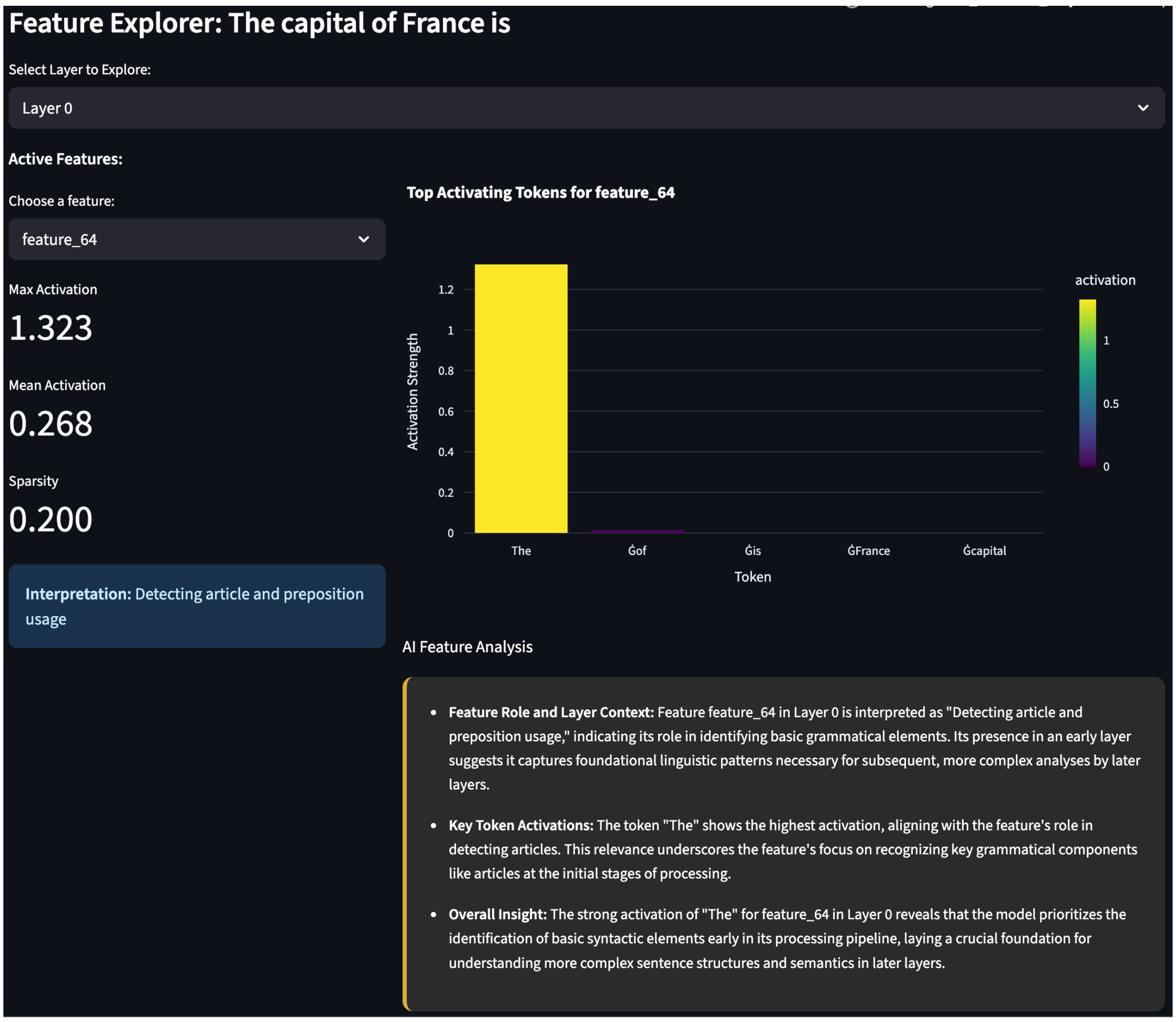
: [███] (0.34)

`The`

: [ ] (0.09)

This shows what parts of the prompt the subnetwork is 'paying attention to.'

Figure 12: Circuit Trace Analysis: Subnetwork Explorer. This interactive tool allows users to isolate and visualize the specific computational pathways connected to a chosen feature, revealing its upstream influences and downstream effects.



Feature Explorer: The capital of France is

Select Layer to Explore:

Layer 0

Active Features:

Choose a feature:

feature_64

Max Activation

1.323

Mean Activation

0.268

Sparsity

0.200

Interpretation: Detecting article and preposition usage

Top Activating Tokens for feature_64

AI Feature Analysis

- **Feature Role and Layer Context:** Feature feature_64 in Layer 0 is interpreted as "Detecting article and preposition usage," indicating its role in identifying basic grammatical elements. Its presence in an early layer suggests it captures foundational linguistic patterns necessary for subsequent, more complex analyses by later layers.
- **Key Token Activations:** The token "The" shows the highest activation, aligning with the feature's role in detecting articles. This relevance underscores the feature's focus on recognizing key grammatical components like articles at the initial stages of processing.
- **Overall Insight:** The strong activation of "The" for feature_64 in Layer 0 reveals that the model prioritizes the identification of basic syntactic elements early in its processing pipeline, laying a crucial foundation for understanding more complex sentence structures and semantics in later layers.

Figure 13: Circuit Trace Analysis: Feature Explorer. Users can inspect individual features in detail, viewing their top activating tokens, sparsity statistics, and AI-generated functional interpretations.