Semantic Pathway: An Interactive Visualization of Hidden States and Token Influence in LLMs

Mithilesh Kumar Singh* Klaus Mueller†

Department of Computer Science
Stony Brook University

Figure 1: Prior tokens are listed on the left with font size scaled by their attention influence on the selected output token ("city"). Each colored trajectory represents a semantic pathway, showing how its hidden state evolves across layers. Faded red attention arcs highlight influential tokens at layer L6, with the arc thickness indicating attention weight. Hovering over a node reveals its top incoming contributors. The radial output view displays the predicted token ("city") alongside top alternatives.

Abstract

Transformer-based language models have demonstrated remarkable capabilities across various tasks, yet their internal mechanisms—such as layered representations, distributed attention, and evolving token semantics—remain challenging to interpret. We present *Semantic Pathway*, an interactive visual analytics tool designed to reveal how token representations evolve across layers in autoregressive Transformer models such as GPT-2. The system integrates layerwise semantic trajectories, attention overlays, and output probability views into a unified interface, enabling users to trace how meaning accumulates and decisions emerge during generation. To reduce visual and interaction complexity, Semantic Pathway incorporates attention-based influence filtering, optional nearest-token projections, and a Compare Mode for analyzing divergence across alternate outputs. The design prioritizes interpretability and usability, supporting both fine-grained inspection and high-level exploration of sequence modeling behavior. This work contributes to ongoing efforts to make language models more interpretable, educationally accessible, and open to diagnostic insight.

Index Terms: Large Language Models, Interpretability, Semantic Pathways, Attention Visualization, Interactive Visualization

*e-mail: mkssingh@cs.stonybrook.edu
†e-mail: mueller@cs.stonybrook.edu

1 Introduction

With the rapid growth and widespread adoption of large language models (LLMs) in critical domains, gaining a deeper and more structured understanding of their internal mechanisms has become increasingly important [3, 5]. Improving transparency into how these models build and transform semantic meaning during generation can foster greater trust, support explainability efforts, and enable a more responsible deployment. However, despite growing efforts in model interpretability, including initiatives like Transformer Circuits [5], the internal dynamics through which autoregressive models such as the GPT family, LLaMA, Mistral, and others develop and transform semantic representations, which are crucial for trustworthy and explainable behavior, remain challenging to trace and understand, especially from a human-in-the-loop perspective where intermediate reasoning steps are often opaque to users.

Understanding these internal dynamics is challenging due to the architectural complexity of transformer models. The representation of each token is shaped through a sequence of non-linear transformations in multiple layers [13, 15], each containing high-dimensional hidden states and multiple attention heads [17]. This makes it difficult to form a cohesive view of how semantic meaning emerges and evolves during generation. Prior interpretability tools have addressed parts of this challenge: the Illustrated Transformer [2] provides intuitive attention explanations; ExBERT [8] and LIT [16] enable interactive exploration of token embeddings; VisBERT [1] visualizes hidden state activations at individual layers for encoder-based models; and LogitLens4LLMs [19] projects hidden states through the output layer to reveal intermediate token predictions in decoder models.

While these approaches reveal valuable patterns, they typically

isolate a single model component, such as attention weights, static embeddings, or output logits, and rarely provide a unified view of how semantic meaning is progressively constructed during generation [6, 10]. Recent work has emphasized the need for more holistic and usable interpretability strategies that go beyond surface-level attention maps, particularly for language models used in high-stakes or iterative contexts [20]. In parallel, visualization systems developed within the visualization research community have highlighted the importance of interactive and multiscale representations [7], demonstrating how layered internal behavior can be made tractable through summarization and exploration. Seq2Seq-Vis [14], for instance, visualizes encoder-decoder alignments and attention dynamics in sequence-to-sequence models, but is not designed for tracing semantic drift or evolving representations within autoregressive decoders. AttentionViz [21] presents a global view of attention patterns by embedding queries and keys jointly across sequences, and CommonsenseVIS [18] provides insight into the reasoning behavior in LLMs. However, these systems generally focus on encoder-based models or offer static summaries and are not designed to trace semantic transformations across layers during autoregressive generation. This gap motivates the need for interactive systems that unify attention, hidden states, and token predictions into a coherent, evolving semantic narrative.

Building on these insights, we introduce *Semantic Pathway*, a visualization framework designed to trace the development of semantic meaning through the intermediate representations of autoregressive Transformer models. Rather than examining attention or hidden states in isolation, Semantic Pathway provides an integrated view of how semantic structure evolves across all layers during token generation. To support detailed exploration and manageable analysis, we focus on GPT-2 for initial development due to its moderate size (12 layers and 12 attention heads) [12].

Semantic Pathway unifies multiple facets of model behavior into a single interactive exploration. It visualizes hidden states to show how token representations transform across layers, attention-based association arcs to reveal internal focus patterns, and influence projections to approximate how prior tokens contribute to prediction. Although attention is not a causal explanation [9], it serves as a valuable signal of contextual influence and routing. By combining attention influence with semantic trajectories, the system enables users to explore how context influences token generation and how different semantic pathways emerge in response to surrounding text.

Through fine-grained semantic analysis and multiscale visualizations, Semantic Pathway offers an educational and exploratory lens into model behavior. While acknowledging the complexity and opacity of large models, our goal is to provide a more accessible and interpretable view of how semantic reasoning unfolds internally, thereby contributing to broader efforts in developing transparent and trustworthy AI systems.

2 System Overview

Semantic Pathway is an interactive visualization system for exploring how semantic representations evolve layer by layer within autoregressive Transformer models. It integrates model outputs such as hidden states, attention weights, and logits into a coordinated visual environment that supports tracing token-level dynamics during generation. The system emphasizes both structure and interactivity, allowing users to follow how context shapes the semantic trajectory of each token, observe internal focus patterns, and examine how semantic pathways emerge from earlier tokens. While initial development targets GPT2 for its manageable size, the framework supports other autoregressive models and adapts visualizations to highlight the most informative transformations. Model outputs are extracted during forward passes and structured into formats that drive the visual components and interactions.

2.1 Data Processing Pipeline

For each generated token, the system collects hidden states, attention weights, and output logits across all layers for all prior tokens in the sequence. These internal representations form the foundation of the visual encodings in *Semantic Pathway*.

Each hidden state is a high-dimensional vector ($d = 768$ for GPT-2). To visualize how token meaning evolves across layers, we project these hidden states into a two-dimensional space using t-SNE. The resulting 2D coordinates define the semantic trajectory of a token through the model. To provide interpretability, we also compute the top-K nearest tokens in the embedding space for each hidden state, anchoring it to semantically similar vocabulary terms.

Attention weights are extracted as square matrices of shape $[\text{seq\_len}, \text{seq\_len}]$ at each layer. Since Transformer architectures use multi-head attention (12 heads in GPT-2), we average across heads to produce a single aggregated matrix per layer. These scores are used to draw directed arcs that visualize attention-based influence from all prior tokens to the token selected for pathway rendering, at any selected layer.

For each generated token t_g, we compute an attention-based influence score for every prior token t_i by averaging the attention received across heads at each layer and summing across layers:

$$\text{Influence}(t_i) = \sum_{\ell=1}^{L} \frac{1}{H} \sum_{h=1}^{H} \text{Attn}^{(\ell,h)}(t_g, t_i)$$

These scores determine which prior tokens appear in the left panel and scale their font size, emphasizing the most relevant contributors for pathway exploration.

Finally, output logits are recorded for each generated token after the final layer. Applying softmax to these logits yields a probability distribution over the vocabulary. We extract the top-N most likely tokens, allowing users to inspect alternative predictions and assess model uncertainty.

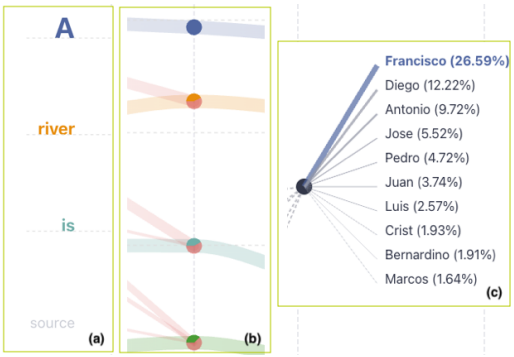

Figure 3: (a) Prior tokens with font size scaled by attention influence. (b) Pathway slice with attention arcs and pie charts showing incoming attention—self-attention in dark (pathway color), others in red. (c) Output view with top predicted token and softmax alternatives.

2.2 Semantic Pathway Views

The system interface begins with a prompt input and model selection (e.g., GPT-2). Upon initiating generation, the model produces tokens one by one, conditioned on both the prompt and previously generated tokens. The resulting sequence includes both input tokens (used as context) and newly generated tokens, all displayed in a horizontal strip. Clicking a generated token triggers the visualization of its internal computation.

The left panel displays a filtered set of prior tokens selected based on their attention-based influence on the selected output token (see Fig. 3, part (a)). Font size is scaled proportionally to in-

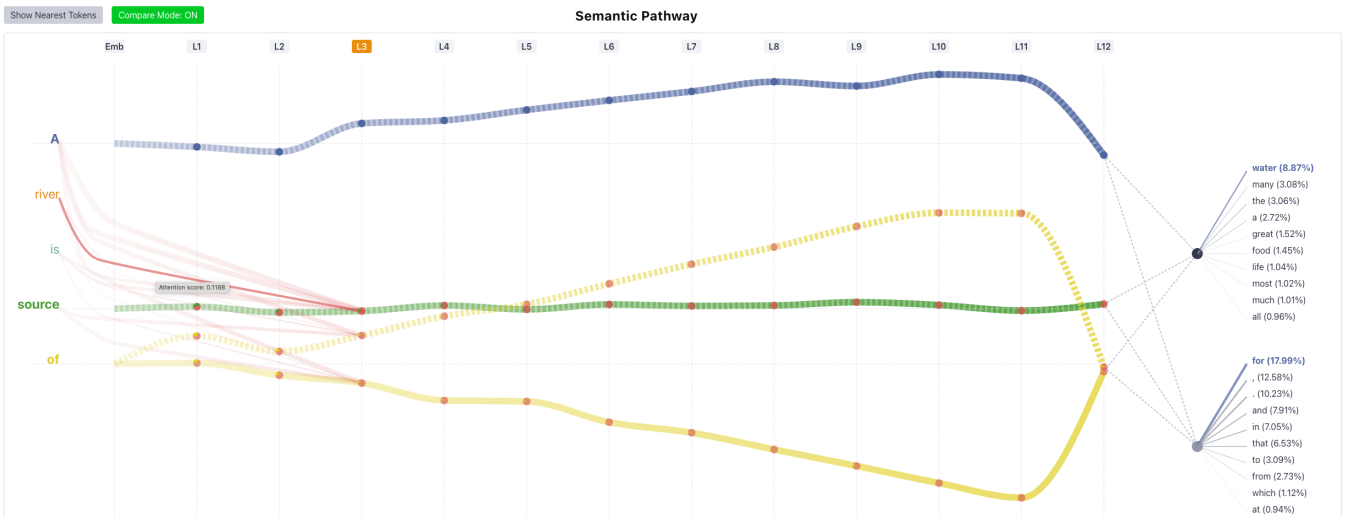

Figure 2: Compare Mode showing diverging semantic pathways for "of" across two outputs ("water" and "for"). Although positionally identical, the token's hidden states diverge due to recomputation at each generation step. Solid and dotted lines represent the two continuations.

fluence, helping users quickly identify the most relevant context. Clicking on any of these tokens reveals its semantic pathway.

Each semantic pathway consists of a series of nodes arranged in layers, illustrating how the internal representation of a token evolves through these layers. The first node appears at the embedding layer, adjacent to the token text, establishing its relative origin. Subsequent nodes trace the transformation of hidden states across layers, projected into two-dimensional space using t-SNE. Each pathway is color-coded based on the selected token, with dashed lines indicating the second token in Compare Mode and a dashed vertical line visually anchors the origin. The number of layers and corresponding nodes are derived directly from the shape of the hidden state tensor. This pathway provides a visual representation of semantic drift, illustrating how a token's meaning evolves as it contributes to the selected output token.

To support semantic interpretability, each hidden state is compared to the model's embedding space to retrieve the top-K nearest tokens by cosine similarity. These nearest tokens serve as optional semantic anchors and are shown using the *Show Nearest Tokens* toggle (see Fig. 1). This option is disabled at the final layer, where representations are projected into the output space and proximity to embeddings becomes uninformative.

Attention arcs provide additional context by showing how all prior tokens influence the pathway token at a specific layer. When users click a layer label, red arcs are rendered from each prior token to the pathway token, with thickness scaled to attention weight. These attentions are averaged across heads for clarity. Although not causal, they provide interpretable signals of contextual focus and the distribution of influence.

Each node in the semantic pathway includes a pie chart that summarizes attention received from prior tokens at that layer, with a distinct segment for self-attention. The self-attention slice is filled with a dark shade matching the pathway color, while attention from other tokens is shown in red (see Fig. 3, part (b)). Hovering over a node reveals a ranked list of top contributors and their corresponding attention percentages, allowing for fine-grained inspection without overwhelming the interface.

Finally, the model's output distribution is displayed as a radial plot after the last layer. The predicted token is centered, with top-N alternatives shown around it. Each line's length reflects softmax probability, helping users assess model uncertainty and plausible continuations (see Fig. 3, part (c)).

Compare mode allows the simultaneous selection of two generated tokens. Their pathways are rendered together using solid and dashed lines, enabling users to compare how the same prior token may evolve differently. Attention arcs and output views are also shown side by side, supporting contrastive analysis of model behavior (Fig. 2). The interface preserves user state across interactions, such as selected layers and Compare Mode toggles.

3 Observations

Using Semantic Pathway to explore hidden states, attention patterns, and output predictions, we identified several recurring behaviors in how Transformer models internally process language. These insights emerged across diverse prompts and generation scenarios.

In early layers, hidden states change gradually as tokens begin to form internal structure. From middle to deeper layers, representations drift more noticeably, capturing richer context. At the final layer, trajectories sharply turn or collapse inward, signaling compression into the output space. This progression reflects a shift from incremental encoding to decisive semantic shaping, and finally, predictive resolution.

Attention patterns also evolve across layers. Early attention is diffuse and dominated by self-attention. As layers deepen, the model refines its focus, reinforcing key tokens while others fade. Self-attention declines, and attention becomes increasingly focused on a few influential tokens. Positional decay is also evident—nearby tokens typically receive more attention than distant ones. Yet, the model compensates for this by reintroducing specific tokens over long ranges in later layers. The first input token often remains a strong influence throughout many steps, serving as a contextual anchor.

Nearest-token projections provide a semantic lens into hidden states. In intermediate layers, nearest neighbors are often consistent function words like "of" or "in." Although semantically light, their layerwise stability suggests an interpretable structure. In contrast, final layer neighbors are erratic or repeated across contexts, e.g., "SPONSORED" or "Reviewer", indicating that this space prioritizes output optimization over meaning and no longer supports useful semantic anchoring.

Compare Mode reveals how the same prior token can evolve differently depending on the output. Since hidden states are recomputed at each generation step, prior tokens shift subtly as the context grows, even without positional changes. Some maintain similar

semantic pathways, while others diverge sharply in deeper layers (Fig. 2). Notably, these divergences occur even when attention arcs are nearly identical, underscoring that semantic representations are shaped not just by attention but by the full transformation stack applied at each step.

Insight: Final-layer hidden states often lose semantic clarity as they collapse into logit space. Interpretability is strongest mid-depth, where semantic structure evolves most meaningfully.

4 Challenges and Design Decisions

Visualizing many layers in deep models quickly exceeds available screen width and introduces visual clutter. Even when space permits, showing every layer often obscures meaningful transitions due to cognitive overload. **Design Decision:** We render semantic trajectories using 2D projections and display only a subset of layers based on semantic deviation. This emphasizes informative transitions without burdening users with redundant or minor changes. Attention overlays can be toggled per layer to inspect internal focus without introducing clutter.

Hidden states are high-dimensional vectors that change across layers, making them difficult to interpret directly. **Design Decision:** We apply t-SNE to project hidden states into 2D space, enabling semantic comparison across layers. While PCA was initially explored, t-SNE better preserved the local structure of the data. To interpret the projection, we overlay the nearest token from the embedding space. Since these often include semantically light terms (e.g., "the", "in"), a toggle allows users to hide them, reducing visual noise.

Each generated token attends to all prior tokens (T) across multiple heads (H) and layers (L), forming a dense attention tensor of shape $[L \times H \times T \times T]$. Identifying the most influential context tokens requires careful aggregation. **Design Decision:** We average the attention weights across heads, then sum across layers to compute context token influence. These influence scores determine which prior tokens appear in the left panel and scale their font size, emphasizing the most relevant contributors for pathway exploration.

Compare Mode introduces complexity by displaying multiple semantic pathways and associated views. Overlapping arcs and diverging trajectories can create visual confusion. **Design Decision:** We limit comparison to two output tokens, distinguish their pathways using solid and dashed lines, and synchronize all views to maintain clarity.

As users switch between output tokens, the set of influential prior tokens can change, potentially causing disorientation in users. **Design Decision:** We maintain original token ordering in the left panel and persist highlight states for previously selected tokens, ensuring interaction stability.

Users differ in their level of interest in exploring model behavior. While experts may seek fine-grained inspection, others may prefer high-level overviews. **Design Decision:** We support multi-level exploration through semantic pathway summaries, attention arc overlays, and hover-based breakdowns, enabling flexible yet interpretable analysis for diverse user needs.

5 Limitations

Semantic Pathway currently targets smaller decoder-only models (e.g., GPT-2) and has not been evaluated on deeper or instruction-tuned architectures. While effective for short and medium-length sequences, the interface may become visually cluttered when applied to longer contexts or document-scale generations. Compare Mode is limited to two output tokens, which restricts temporal or multi-token exploration.

Hidden states are projected using t-SNE, which may introduce spatial distortions; distances and trajectories should be interpreted qualitatively. We average attention weights across heads, potentially suppressing head-specific behaviors. Influence scores are computed heuristically and may not reflect causal attribution. While the system supports exploratory insight into token-level dynamics, it does not provide definitive explanations. Semantic Pathway does not incorporate gradient-based attribution techniques [4] or circuit-level tracing frameworks like TransformerLens [11], which could offer complementary, causally grounded perspectives on internal model behavior.

6 Future Work

Semantic Pathway could be extended to support larger or instruction-tuned models such as GPT-3 or LLaMA-2, which feature deeper architectures and more complex prompting behavior. These models may require scalable summarization methods and new abstractions to manage their expanded internal structure.

To handle more extensive sequences and document-scale generations, future designs may incorporate features such as zooming, panning, or collapsible views to support hierarchical navigation.

Compare Mode may evolve beyond pairwise outputs to support top-k continuations or temporally aligned comparisons, enabling richer exploration of divergence and ambiguity in generation.

Deeper semantic insight could be gained by clustering hidden states across tokens and layers to reveal functional units such as syntactic roles or semantic detectors. Comparing hidden states with the vocabulary embedding space may also clarify how intermediate layers refine or repurpose static embeddings. These methods may also help connect internal dynamics to interpretable features, such as syntax, sentiment, or factual grounding.

Future work could explore per-head attention analysis, attention flow tracking, or clustering of attention patterns to uncover head specialization and routing behavior.

Finally, aligning internal signals with user-defined semantics, task labels, or prediction outcomes could bridge model behavior with external meaning, supporting both interpretability and downstream diagnostics.

7 Conclusion

Semantic Pathway introduces an interactive framework for exploring how token representations evolve across layers in autoregressive language models. By integrating hidden state projections, attention-based influence visualizations, and comparative pathway analysis, the system reveals layered dynamics of meaning construction and context shaping during generation.

Rather than offering absolute interpretability, the system emphasizes structured transparency, surfacing patterns that support human reasoning and diagnostic insight. Observations reveal persistent influence from early tokens, shifting semantic pathways across layers, and reduced interpretability near the output stage.

While the current system focuses on autoregressive architectures and short sequences, it lays the foundation for future extensions toward deeper models, longer contexts, and user-aligned signals. Attention-based influence scores, although heuristic, provide a structured lens for tracing semantic evolution and contextual influence. Semantic Pathway supports both exploration and educational insight for technical and non-technical audiences. Beyond research applications, it may also serve as a pedagogical tool, offering learners a visual scaffold to understand how language models construct meaning layer by layer.

This layered progression can be interpreted as a narrative of competing possibilities. Early layers maintain broader semantic ambiguity, with multiple plausible continuations coexisting—for instance, "life" or "food" as alternatives to "water" in the prompt "A river is source of…". As the model advances, attention and hidden state transformations incrementally filter and refine these candidates. By the final layer, semantic uncertainty collapses into a

confident prediction, highlighting how the model organizes meaning over depth as a form of internal time.

Supplemental Materials

A short video demonstration of the Semantic Pathway interface is available at https://vimeo.com/1080448380/2f38ba5dfc. It showcases key interactions and visual components discussed in the paper. The video is accessible via direct link only.

References

[1] B. v. Aken, B. Winter, A. Löser, and F. A. Gers. Visbert: Hidden-state visualizations for transformers. In *Companion Proceedings of the Web Conference 2020*, pp. 207–211, 2020. 1

[2] J. Alammar. The illustrated transformer. http://jalammar.github.io/illustrated-transformer/, 2018. Accessed: 2025-04-28. 1

[3] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1

[4] H. Chefer, S. Gur, and L. Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 782–791, 2021. 4

[5] N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021. 1

[6] M. Geva, R. Schuster, J. Berant, and O. Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020. 2

[7] F. Hohman, H. Park, C. Robinson, and D. H. P. Chau. S ummit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. *IEEE transactions on visualization and computer graphics*, 26(1):1096–1106, 2019. 2

[8] B. Hoover, H. Strobelt, and S. Gehrmann. exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276*, 2019. 1

[9] S. Jain and B. C. Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019. 2

[10] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*, 2019. 2

[11] N. Nanda and J. Bloom. Transformerlens. https://github.com/neelnanda-io/TransformerLens, 2022. Accessed: 2025-04-30. 4

[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[13] A. Rogers, O. Kovaleva, and A. Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the association for computational linguistics*, 8:842–866, 2021. 1

[14] H. Strobelt, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush. S eq 2s eq-v is: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics*, 25(1):353–363, 2018. 2

[15] I. Tenney, D. Das, and E. Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019. 1

[16] I. Tenney, J. Wexler, J. Bastings, T. Bolukbasi, A. Coenen, S. Gehrmann, E. Jiang, M. Pushkarna, C. Radebaugh, E. Reif, et al. The language interpretability tool: Extensible, interactive visualizations and analysis for nlp models. *arXiv preprint arXiv:2008.05122*, 2020. 1

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[18] X. Wang, R. Huang, Z. Jin, T. Fang, and H. Qu. Commonsensevis: Visualizing and understanding commonsense reasoning capabilities of natural language models. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):273–283, 2023. 2

[19] Z. Wang. Logitlens4llms: Extending logit lens analysis to modern large language models. *arXiv preprint arXiv:2503.11667*, 2025. 1

[20] X. Wu, H. Zhao, Y. Zhu, Y. Shi, F. Yang, T. Liu, X. Zhai, W. Yao, J. Li, M. Du, et al. Usable xai: 10 strategies towards exploiting explainability in the llm era. *arXiv preprint arXiv:2403.08946*, 2024. 2

[21] C. Yeh, Y. Chen, A. Wu, C. Chen, F. Viégas, and M. Wattenberg. Attentionviz: A global view of transformer attention. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):262–272, 2023. 2