Mechanistic Interpretability of Large Language Models: A Survey

Pradyumna Shyama Prasad

ABSTRACT

This survey provides a comprehensive overview of mechanistic interpretability in large language models (LLMs), focusing on techniques to understand the internal computations and decision-making processes of neural networks, transformers, and other complex architectures. We introduce key concepts such as features, polysemanticity, and circuits, which form the foundation for understanding model behavior at a granular level. The paper explores a taxonomy of approaches, categorizing them into observational techniques like probing and sparse autoencoders, and interventional techniques such as ablation and activation patching. We review recent advances in these areas, discussing how they contribute to uncovering the underlying mechanisms of LLMs.

We address the challenges of scaling interpretability techniques to larger and more complex models, highlighting issues such as computational feasibility and the difficulty of disentangling polysemantic features. We examine the current limitations of interpretability methods and their implications for AI transparency and safety. Furthermore, we identify open questions and potential future directions for research, including the development of more robust evaluation metrics and approaches for generalizing interpretability across different tasks and model architectures.

KEYWORDS

Mechanistic interpretability, deep learning, neural networks, transformers, interpretability, AI, challenges, future directions

1 INTRODUCTION

1.1 Background and Motivation

Since the advent of the Transformer (Vaswani et al., 2017), language models have achieved near-human abilities on various tasks, from mathematics to programming (Dubey et al., 2024, OpenAI et al., 2024, Qwen, 2024). However, their interpretability has not kept pace with their capabilities, posing challenges for safety, transparency, and trust in high-stakes applications.

While existing interpretability methods focus on input-output relationships and high-level summaries (Bolukbasi et al., 2021, Zhao et al., 2023), mechanistic interpretability aims to reverse-engineer neural networks, making their internal behavior understandable (Olah et al., 2020). This approach is crucial for debugging models, identifying biases, and ensuring reliable and ethical AI systems (Räuker et al., 2023).

The need for mechanistic interpretability has become increasingly urgent as language models grow in size and capability. As Bereska and Gavves (2024) argue, understanding the internal mechanisms of these models is essential for ensuring their alignment with human values and avoiding potential catastrophic outcomes. Mechanistic interpretability offers several key benefits. Firstly, it enhances safety and alignment by providing insights into model decision-making processes, which can help identify and mitigate potential safety risks, such as deceptive or misaligned behaviors (Bereska and Gavves, 2024). Secondly, it improves robustness and generalization; understanding internal model mechanisms can lead to improvements in these areas, as highlighted by Rai et al. (2024). Lastly, it enables targeted improvements, as mechanistic insights allow for more precise and effective model modifications, enabling researchers to enhance specific capabilities or correct undesired behaviors (Rai et al., 2024).

However, mechanistic interpretability faces significant challenges with modern deep learning architectures. Large language models like GPT-3, with 175 billion parameters, are difficult to interpret due to their size and complexity, with individual neurons often representing multiple, overlapping features. Moreover, the field remains pre-paradigmatic, lacking widely accepted definitions for many key concepts (Bereska and Gavves, 2024).

Despite these challenges, the potential benefits of mechanistic interpretability make it a crucial area of research as we continue to develop and deploy increasingly powerful AI systems. This survey aims to provide a comprehensive overview of current mechanistic interpretability techniques, their applications, and the challenges they face in the context of large language models.

1.2 Scope of the Survey

This survey focuses on mechanistic interpretability in large language models (LLMs), aiming to uncover and explain their internal structures and computations. Unlike general interpretability methods that emphasize input-output mappings, we explore techniques that investigate the internal workings of these models. Key aspects covered in this survey include feature representation across LLM layers, circuits and modular computation within models, and the challenges posed by polysemanticity and superposition. We also discuss interventional techniques for causal analysis, as well as the scalability of interpretability methods to models with billions of parameters.

1.3 Contributions of This Survey

Recent surveys have significantly advanced the field of mechanistic interpretability (MI). Notably, Rai et al. (2024) provided a practical review focusing on techniques and applications in transformer-based models, while Bereska and Gavves (2024) offered an AI safety-focused analysis of MI across various AI systems. Our work complements these surveys and offers distinct contributions:

(1) **Comparative Overview:**
- Rai et al. (2024): A practical guide for beginners, with detailed coverage of MI techniques and their implementations in transformer models.
- Bereska and Gavves (2024): A broader analysis of MI, emphasizing AI safety across different systems beyond language models.
- Our survey: Bridges these approaches by focusing specifically on large language models (LLMs), offering both practical and theoretical insights.

(2) **In-depth Analysis of Sparse Autoencoders (SAEs):** Previous surveys mention SAEs, but our work provides a detailed examination of SAEs in the context of LLM interpretability. We explore their theoretical foundations, practical implementations, and potential to disentangle polysemantic features in LLMs, offering a more comprehensive analysis of this emerging technique.

(3) **Focus on Intervention Techniques:** Our survey reviews interventional methods, including activation patching, causal tracing, and causal scrubbing, synthesizing recent developments and highlighting their strengths, limitations, and applications for understanding LLM internals. This complements the broader technique coverage in Rai et al. (2024) and the safety implications discussed in Bereska and Gavves (2024).

By providing specific insights and synthesizing practical and theoretical perspectives, our survey offers a focused yet comprehensive view of key frontiers in mechanistic interpretability for large language models. This complements existing literature by providing a specialized perspective on emerging techniques and their applications in understanding LLM internals.

2 KEY CONCEPTS AND DEFINITIONS

2.1 Features are the Fundamental Units of Representation

Features are the fundamental units of representation in neural networks, encapsulating specific patterns or properties of input data. They are encoded within neuron activations across different layers, forming the building blocks of how a model processes and understands data. In transformer-based models, features are represented in layers of attention heads, feed-forward networks, and residual streams.

A **feature** can be thought of as a specific input property, such as recognizing a linguistic pattern or detecting an edge in an image. For example, researchers found a neuron that activates only for French text, demonstrating a clear correspondence between the neuron and a specific feature (Gurnee et al., 2023).

2.1.1 Polysemanticity. It is not always the case that one neuron will correspond to a single feature. Many neurons in deep learning models exhibit **polysemanticity**, meaning they activate in response to multiple unrelated features. For instance, a neuron may fire when processing both "French text" and "Base64-encoded text," or respond to different concepts in distinct contexts. For example, out of over 300,000 neurons in GPT-2, researchers at OpenAI were able to find explanations that explained over 50% of the variance for only about five thousand neurons (only 1.6% of them) (Bills et al., 2023, C, 2024).

One hypothesis for why this occurs because neural networks often need to represent more features than there are available neurons (Elhage et al., 2022). As a result, some neurons encode multiple features simultaneously, leading to complex activation patterns that do not correspond to just one interpretable concept. Understanding polysemanticity is a major challenge for neural networks because they exponentially increase the number of combinations of neurons needed to understand a feature. For example, if there is one neuron with five different possible features, which connects to another neuron with five other different features, that is 25 different combinations that need to be considered. (Olah et al., 2020)

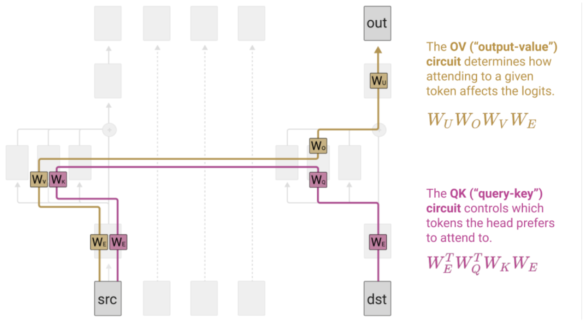
out
src
dst
W_U
W_O
W_V
W_K
W_Q
W_E
W_E
W_E
The OV ("output-value") circuit determines how attending to a given token affects the logits.
$W_U W_O W_V W_E$
The QK ("query-key") circuit controls which tokens the head prefers to attend to.
$W_E^T W_Q^T W_K W_E$

Figure 1: An example of a circuit in a transformer model. From (Olah et al., 2020)

2.1.2 Superposition. One hypothesized cause for polysemanticity is **superposition**. As explained above, superposition is when a neural network represents more features than it has dimensions. Superposition occurs when a neural network represents more features than the number of available neurons (or dimensions) in its layers. In other words, the network must "compress" multiple features into the same set of neurons. This leads to a situation where individual neurons activate for multiple, potentially unrelated features, contributing to polysemanticity.

Superposition can be thought of as a method for the neural network to efficiently store and process a large number of features with a limited number of neurons. Instead of dedicating a neuron to a single feature, the network encodes several features into overlapping directions in the activation space. While this allows the model to represent more information with fewer resources, it also makes it harder to interpret because each neuron is no longer aligned with a distinct, interpretable feature.

2.2 Circuits

A circuit is a subgraph within a neural network connecting neurons or features across layers, forming a coherent, interpretable mechanism. Circuits reveal how individual neurons contribute to higher-level functions, providing insights into specific computations and feature detection patterns.

2.2.1 Circuits in Transformers. In transformers, circuits are pathways through which attention heads, tokens, and layers interact to compute functions like attention patterns and predictions. The *residual stream* enables information flow, creating a modular and compositional structure.

2.2.2 Attention Heads and Circuits. Attention heads act as subcircuits, facilitating information transfer between tokens through the **query-key-value (QKV)** mechanism:

- **QK Circuits** govern which tokens attend to each other.
- **OV Circuits** determine a token's influence on final predictions.

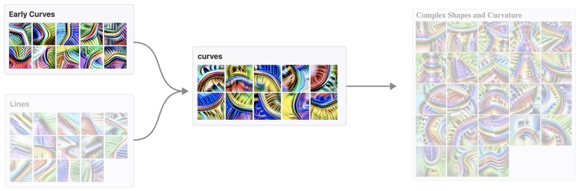

Figure 2: An example of a circuit in a CNN model. From (Olah et al., 2020)

2.2.3 Residual Stream and Information Flow. The residual stream acts as the communication backbone for circuits in transformers, enabling interaction between attention heads and layers. This allows for **higher-order circuits**, forming sophisticated representations and complex dependencies between tokens.

3 TAXONOMY OF APPROACHES

We draw on two previously existing reviews to classify approaches in mechanistic interpretability (Bereska and Gavves, 2024, Rai et al., 2024). Based on these existing frameworks, we classify techniques in mechanistic interpretability into two types: **observational techniques** and **interventional techniques**.

3.1 Observational Techniques

3.1.1 Probing. Probing is a diagnostic technique used to reverse engineer the internal representations of neural networks, particularly in NLP. It involves training simple classifiers (probes) to predict specific properties from the activations of hidden layers or individual neurons. These properties can range from linguistic features to abstract information like spatial and temporal relationships.

Probing offers insight into the latent knowledge encoded by neural models, revealing how well internal representations capture targeted properties and how these properties are distributed across neurons and layers (Bolukbasi et al., 2021, Zhao et al., 2023). This helps in understanding how neural networks encode and process information at various levels of abstraction.

3.1.2 Variations within Probing.

Linear Probing: The most common form, using simple linear classifiers to predict properties. Gurnee and Tegmark (2024) used linear ridge regression to predict spatial-temporal information, while Marks and Tegmark (2024) classified true and false statements, finding distinct linear directions in the model's representation space.

Nonlinear Probing: Uses more complex classifiers like MLPs to capture nonlinear relationships. Gurnee and Tegmark (2024) found minimal improvements over linear probes for spatial-temporal features, highlighting concerns about overfitting and reduced interpretability.

Sparse Probing: Introduces sparsity constraints to identify specific neurons encoding target features. Gurnee et al. (2023) demonstrated its effectiveness in analyzing large language models, uncovering key neurons for detecting specific features like language type or code structure.

Sparse probing helps address the challenge of **polysemanticity**, where neurons activate for multiple, often unrelated features. By focusing on a smaller set of neurons, it reduces interference from polysemantic neurons and offers a clearer view of feature representation (Olah et al., 2020).

3.1.3 Limitations of Probing. Recent studies have identified several limitations:

- Probe quality doesn't necessarily indicate neuron importance. Antverg and Belinkov (2022) argue that a well-performing probe may still rank neurons inaccurately.
- Distinction between encoded and actively used information. Identifying neurons that encode information doesn't guarantee their use in model predictions (Antverg and Belinkov, 2022).
- Lack of causal interventions to determine essential neurons for model decisions (Elhage et al., 2022).

These limitations highlight the need for caution when interpreting probing results and the importance of complementary techniques in mechanistic interpretability (Bereska and Gavves, 2024, Rai et al., 2024).

Study	Linear	Nonlinear	Sparse	Other
Gurnee et al. (2024)	✓	✓		
Marks et al. (2024)	✓			
Gurnee et al. (2023)			✓	
Antverg et al. (2022)				✓

Table 1: Types of Probing Techniques Used in Recent Studies

3.1.4 Sparse Autoencoders. Sparse autoencoders (SAEs) have emerged as a powerful tool for addressing the challenges of interpreting large language models, particularly the issue of polysemanticity. While traditional probing methods struggle with causal clarity and the complexity of these models, SAEs offer a novel approach by encouraging sparsity in neural representations.

SAEs are a type of autoencoder that enforces sparsity in its latent space representation. Unlike traditional autoencoders, which compress input data into a dense latent representation, SAEs ensure that only a small number of neurons in the hidden layer are activated for any given input. This is typically achieved by adding a regularization term to the loss function that penalizes dense activations. The goal is to force the network to learn more efficient, distinct features by reducing redundancy and ensuring that each neuron in the hidden layer captures a unique aspect of the data (Bricken et al., 2023, Cunningham et al., 2023).

The key advantage of SAEs in interpreting large language models is their ability to mitigate polysemanticity. By activating only a subset of neurons for each input, SAEs encourage each neuron to represent a distinct and interpretable feature, rather than encoding multiple overlapping or unrelated features. This makes it easier to trace connections between neurons and specific concepts, leading to more interpretable and monosemantic representations.

Advantages of SAEs. Recent work by Bricken et al. (2023) demonstrated the power of SAEs in learning disentangled features from transformer models. Using a large dataset of 8 billion data points, they decomposed MLP layer activations into sparse, monosemantic features responding to specific contexts such as Arabic script, DNA sequences, base64 strings, and Hebrew script. These features exhibited **high specificity** and **sensitivity** for the contexts they represented, demonstrating the ability of SAEs to produce clearer and more interpretable features than individual neurons.

Recent Advancements. The use of SAEs is particularly effective when compared to traditional probing methods. While probing often lacks causal clarity and may conflate correlation with causation, SAEs enforce a latent space where each feature is monosemantic. This means that each activation has a clear and interpretable relationship with a specific concept, making it easier to reverse-engineer the decision-making processes of large models by attributing behaviors to distinct learned features (Bricken et al., 2023, Cunningham et al., 2023).

Validation Methods. To validate the reliability and specificity of these features, Bricken et al. (2023) and Cunningham et al. (2023) designed computational proxies that estimate the likelihood of certain contexts being present in the data. By correlating the activations of SAEs with these proxies across a large dataset of over 8 billion tokens, they demonstrated that the features learned by the SAE were indeed highly specific to the target contexts. The sparse autoencoder model consistently activated the correct feature when the context was present, reducing the risk of polysemantic neurons confusing different features.

Scaling to Larger Models. Recent efforts have scaled the SAE framework to larger models, addressing the critical challenge of interpretability in state-of-the-art language models. Lieberum et al. (2024) applied SAEs to Gemma 2 models (ranging from 2B to 27B parameters), enabling a layer-wise application of SAEs across different model architectures. The authors also introduced the *JumpReLU* activation function, which dynamically adjusts the number of active neurons based on the input, further refining the specificity and interpretability of the learned features. This framework allows researchers to conduct detailed circuit analysis and understand how model representations evolve across layers, providing a broader picture of how sparse autoencoders perform in large-scale models.

Limitations and Challenges. However, SAEs are not without limitations and challenges. Chaudhary and Geiger (2024) found that while SAEs could disentangle features in smaller models like GPT-2 Small, their performance often degraded the model's factual knowledge when applied to certain tasks, such as separating a city's country and continent. This highlighted the limitations of SAEs in some tasks, especially in comparison to supervised methods like Distributed Alignment Search (DAS), which performed better at disentangling factual knowledge.

Promise and Potential. Additionally, Makelov et al. (2024) addressed these concerns by evaluating SAEs on principled grounds, noting that while SAEs can provide monosemanticity, their performance is highly sensitive to model architecture and the specific task. In their experiments with the IOI (Indirect Object Identification) task, the SAE models demonstrated reasonable disentanglement of features but still faced challenges with preserving the model's overall performance, revealing that more work is needed to balance feature sparsity with model fidelity.

Universality of SAEs. Despite these challenges, SAEs have shown promise in learning higher-level representations that span multiple neurons, capturing complex, real-world features more effectively than individual neurons. This is significant because individual neurons, particularly in deep models, frequently fail to capture complex, real-world features due to polysemanticity. By spanning multiple neurons, SAEs learn more meaningful and contextually accurate representations, such as identifying languages or specific encoding formats, making the internal mechanics of the model more transparent (Bricken et al., 2023, Cunningham et al., 2023).

Future Directions. Furthermore, SAEs exhibit **universality**, meaning that the learned features remain consistent across multiple runs and different models. This was demonstrated in Bricken et al. (2023), where the same features could be extracted by the SAE across different models, further reinforcing the interpretability and reliability of the learned representations. The public release of these models and their learned features (Lieberum et al., 2024) has made it easier for other researchers to explore and apply SAEs to a wide range of interpretability tasks.

Future research directions for SAEs in mechanistic interpretability include several key areas. **Developing techniques to maintain model performance while increasing feature interpretability** is an important goal, ensuring that interpretability improvements do not come at the cost of model efficacy. **Exploring the application of SAEs to other model architectures beyond transformers** could uncover new insights across different AI systems. Researchers should also focus on **investigating ways to combine SAEs with other interpretability methods** for a more comprehensive understanding of model behavior. **Creating standardized benchmarks for evaluating the quality and utility of SAE-learned features** will help in assessing their effectiveness. Finally, **studying the potential of SAEs in improving model robustness and out-of-distribution generalization** can lead to more reliable AI systems.

Study	Key Contribution	Findings/Implications
Bricken et al. (2023)	Applied SAE to transformer MLP layers	Decomposed activations into monosemantic features (e.g., Arabic script, DNA sequences)
Cunningham et al. (2023)	Validated SAE features with computational proxies	Demonstrated high specificity of learned features
Lieberum et al. (2024)	Scaled SAE to larger models (Gemma 2)	Introduced JumpReLU activation for dynamic neuron activation
Chaudhary et al. (2024)	Evaluated open-source SAEs	Found limitations in certain tasks (e.g., separating city's country and continent)
Makelov et al. (2024)	Principled evaluation of SAEs	Noted sensitivity to model architecture and specific tasks

Table 2: Summary of Sparse Autoencoder (SAE) Studies and Their Contributions

3.2 Interventional Techniques

While observational techniques provide insights into neuron activations, they may only reveal correlations rather than causal relationships. Interventional techniques address this by altering neuron activations to observe changes in model output (Bereska and Gavves, 2024, Rai et al., 2024).

3.2.1 *Ablation techniques.*

Ablation techniques explore causal relationships by systematically removing or deactivating model components and observing the effect on task performance. This helps identify components of computational circuits performing specific tasks, providing causal evidence for their existence (Hanna et al., 2023, Wang et al., 2022).

For example, ablating attention heads in transformers can determine their necessity for tasks like indirect object identification (IOI). A significant performance drop when ablating a Name Mover Head suggests its role in a circuit for copying correct tokens (Wang et al., 2022). Similarly, ablations in GPT-2 identified specific neurons and layers critical for mathematical operations like greater-than comparisons (Hanna et al., 2023).

However, complete component removal may introduce unintended effects, motivating more refined methods like path patching and knockouts.

Path Patching. Path patching allows for more precise analysis of component contributions by selectively intervening in computational pathways. It involves running the model on two inputs and swapping outputs from specific components between them (Goldowsky-Dill et al., 2023, Wang et al., 2022).

This technique has been used to identify critical attention heads in tasks like selecting correct indirect objects in sentences. By patching outputs between inputs, researchers identified Name Mover Heads as crucial for copying correct names to output positions (Wang et al., 2022). Further advancements have demonstrated how path patching can refine hypotheses about circuit structures and quantify their contributions to model outputs (Goldowsky-Dill et al., 2023).

Knockouts. Knockouts offer a less disruptive approach by replacing component outputs with neutral values rather than removing them entirely. This preserves model stability while allowing analysis of a component's role (Wang et al., 2022).

In knockout interventions, outputs are replaced with average activations from a reference dataset. This method has revealed important insights, such as the function of S-Inhibition Heads in preventing incorrect predictions and the existence of backup mechanisms like Backup Name Mover Heads (Wang et al., 2022).

3.2.2 *Best Practices for Activation Patching and Interventions.*

Recent work has established best practices for these techniques (Zhang and Nanda, 2024):

1. Choice of Corruption Method. Earlier methods like Gaussian Noising (GN) often led to out-of-distribution behavior (Goldowsky-Dill et al., 2023, Wang et al., 2022). Symmetric Token Replacement (STR) provides a more reliable alternative by keeping corrupted inputs within the model's learned distribution (Zhang and Nanda, 2024).

2. Evaluation Metrics. While probability difference was commonly used (Wang et al., 2022), logit difference offers a more robust evaluation by directly measuring the patch's impact on decision-making (Hanna et al., 2023, Zhang and Nanda, 2024).

3. Layer and Component Selection. Rather than focusing on single layers or heads, sliding window patching allows for understanding how information flows through multiple components simultaneously (Goldowsky-Dill et al., 2023, Zhang and Nanda, 2024).

4. Handling Circuit Redundancy and Backup Mechanisms. Studies have revealed circuit redundancies where backup mechanisms compensate for disrupted components (Hanna et al., 2023, Wang et al., 2022). Systematic ablation or patching of multiple components is crucial to expose these redundancies (Zhang and Nanda, 2024).

5. Iterative Refinement of Hypotheses. An iterative approach to patching experiments allows for incremental refinement of understanding about circuit structures (Zhang and Nanda, 2024).

These interventional techniques and best practices provide powerful tools for uncovering causal relationships within neural networks, offering deeper insights into the functioning of large language models.

Technique	**Pros**	**Cons**
Ablation	• Simple and effective for identifying critical components • Highlights causal links between components and outputs	• Can cause unintended disruptions in other parts of the model • May miss redundant components
Activation Patching	• Granular control over activations • Helps isolate circuits and trace information flow	• Sensitive to corruption methods used • Interpretation can vary with evaluation metrics
Knockouts	• Preserves model stability • Identifies compensatory mechanisms (backup circuits)	• Requires careful multi-component testing to reveal redundancy • Can be less granular than activation patching
Sliding Window Patching	• Captures interactions across multiple layers • Better for tasks requiring long-range dependencies	• May introduce noise by patching unrelated components • Higher computational complexity

Table 3: Summary of Techniques: Pros and Cons

4 CHALLENGES AND OPEN RESEARCH QUESTIONS

Despite significant advances in mechanistic interpretability, several challenges remain, particularly when applied to large-scale models like GPT-3 and GPT-4. These challenges highlight the limitations of current techniques and point to open questions that future research must address.

4.1 Scalability of Interpretability Techniques

As language models grow in size and complexity, many interpretability methods, including activation patching, ablations, and probing, become computationally expensive and less effective. Techniques developed for smaller models often fail to scale effectively to larger architectures. For example, while activation patching may identify circuits in smaller models like GPT-2, it is significantly more challenging to apply the same technique to models with billions of parameters.

Open Question: How can interpretability techniques be adapted or scaled to handle models with tens or hundreds of billions of parameters? Can we automate the process of identifying circuits in such large-scale models?

4.2 Redundancy and Circuit Complexity

A key challenge in understanding the internal workings of large models is the presence of redundancy within circuits. Often, multiple components (e.g., attention heads or neurons) contribute to the same function, and when one is ablated, others can compensate. This redundancy makes it difficult to isolate which components are truly essential for a particular behavior, complicating the interpretability process.

Open Question: How can we disentangle true causal components from redundant or compensatory mechanisms in neural circuits? What techniques can be developed to systematically identify and understand backup circuits?

4.3 Evaluating Interpretability Methods

A persistent challenge in interpretability research is the lack of standardized evaluation metrics. Many methods rely on subjective criteria, such as human intuition, to assess whether an interpretation is meaningful. However, this approach lacks rigor and can lead to overfitting to simple, human-comprehensible mechanisms without addressing more complex behaviors within the model.

Open Question: What standardized metrics can be developed to rigorously evaluate the effectiveness of interpretability techniques? How can we design evaluation frameworks that reflect the utility of these techniques for practical applications, such as debugging or ensuring AI safety?

4.4 Generalization Across Tasks and Models

Most current research focuses on interpreting specific tasks or models, but the generalizability of these techniques remains unclear. For example, a method that identifies circuits responsible for indirect object identification in one model might not generalize to other tasks, such as mathematical reasoning, or to other models with different architectures.

Open Question: Can we develop interpretability techniques that generalize across different tasks and models? What are the limitations of task-specific techniques, and how can they be addressed to provide broader insights across various domains?

In summary, while current techniques have made progress in mechanistic interpretability, much work remains to be done to address the challenges posed by large models, polysemanticity, and the generalization of techniques. Addressing these open questions is crucial for advancing the field and ensuring that AI systems are transparent, interpretable, and safe.

5 CONCLUSION

Mechanistic interpretability offers a powerful framework for understanding the inner workings of large language models at a granular level, providing insights into how neural networks process information and make predictions. By focusing on internal computations rather than treating models as black boxes, mechanistic interpretability can help researchers and practitioners uncover hidden patterns, identify biases, and ensure the safety and reliability of AI systems.

In this paper, we have reviewed key concepts in the field, including the nature of features, polysemanticity, and circuits, as well as techniques such as probing, ablation, and activation patching. We have highlighted how these approaches provide causal evidence for the existence of circuits in models and discussed the challenges associated with scaling these methods to larger architectures.

Despite the progress made, several open research questions remain, particularly regarding the scalability of interpretability techniques, the disentangling of polysemantic neurons, and the generalization of interpretability methods across different tasks and models. Addressing these challenges is essential for developing robust and reliable interpretability tools that can be applied to the increasingly complex models being used in AI today.

The future of mechanistic interpretability will likely involve the combination of observational and interventional techniques, improvements in evaluation frameworks, and the refinement of methods to handle the complexity of larger models. As we move forward, the continued exploration of these areas will be crucial for ensuring that AI systems are transparent, interpretable, and aligned with human values.

Future work should focus on developing scalable interpretability tools, improving the disentanglement of polysemantic features, and creating standardized evaluation metrics that reflect practical utility. The successful integration of these innovations will pave the way for more interpretable AI systems that can be confidently deployed in real-world applications.

REFERENCES

Omer Antverg and Yonatan Belinkov. 2022. On the Pitfalls of Analyzing Individual Neurons in Language Models. arXiv:2110.07483 [cs.CL] https://arxiv.org/abs/2110.07483

Leonard Bereska and Efstratios Gavves. 2024. Mechanistic Interpretability for AI Safety - A Review. *Transactions on Machine Learning Research* (Aug 2024). https://openreview.net/forum?id=ePUVetPKu6

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. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. 2021. An Interpretability Illusion for BERT. arXiv:2104.07143 [cs.CL] https://arxiv.org/abs/2104.07143

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread* (2023). https://transformer-circuits.pub/2023/monosemantic-features/index.html

Lawrence C. 2024. Superposition is not "just" neuron polysemanticity. https://www.lesswrong.com/posts/8EyCQKuWo6swZpagS/superposition-is-not-just-neuron-polysemanticity. Accessed: 2024-10-12.

Maheep Chaudhary and Atticus Geiger. 2024. Evaluating Open-Source Sparse Autoencoders on Disentangling Factual Knowledge in GPT-2 Small. arXiv:2409.04478 [cs.LG] https://arxiv.org/abs/2409.04478

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse Autoencoders Find Highly Interpretable Features in Language Models. arXiv:2309.08600 [cs.LG] https://arxiv.org/abs/2309.08600

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy Models of Superposition. *Transformer Circuits Thread* (2022). https://transformer-circuits.pub/2022/toy_model/index.html

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing Model Behavior with Path Patching. arXiv:2304.05969 [cs.LG] https://arxiv.org/abs/2304.05969

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding Neurons in a Haystack: Case Studies with Sparse Probing. arXiv:2305.01610 [cs.LG] https://arxiv.org/abs/2305.01610

Wes Gurnee and Max Tegmark. 2024. Language Models Represent Space and Time. arXiv:2310.02207 [cs.LG] https://arxiv.org/abs/2310.02207

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. arXiv:2305.00586 [cs.CL] https://arxiv.org/abs/2305.00586

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2. arXiv:2408.05147 [cs.LG] https://arxiv.org/abs/2408.05147

Aleksandar Makelov, George Lange, and Neel Nanda. 2024. Towards Principled Evaluations of Sparse Autoencoders for Interpretability and Control. arXiv:2405.08366 [cs.LG] https://arxiv.org/abs/2405.08366

Samuel Marks and Max Tegmark. 2024. The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets. arXiv:2310.06824 [cs.AI] https://arxiv.org/abs/2310.06824

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom In: An Introduction to Circuits. *Distill* (March 2020). https://doi.org/10.23915/distill.00024.001

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

Qwen. 2024. Introducing Qwen2-Math. Qwen Blog. https://qwenlm.github.io/blog/qwen2-math/ Accessed: 2024-10-12.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. 2024. A Practical Review of Mechanistic Interpretability for Transformer-Based Language Models. arXiv:2407.02646 [cs.AI] https://arxiv.org/abs/2407.02646

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. 2023. Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks. arXiv:2207.13243 [cs.LG] https://arxiv.org/abs/2207.13243

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Advances in Neural Information Processing Systems 30 (NIPS 2017)* (2017). arXiv:1706.03762 [cs.CL] https://arxiv.org/abs/1706.03762

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 small. arXiv:2211.00593 [cs.LG] https://arxiv.org/abs/2211.00593

Fred Zhang and Neel Nanda. 2024. Towards Best Practices of Activation Patching in Language Models: Metrics and Methods. arXiv:2309.16042 [cs.LG] https://arxiv.org/abs/2309.16042

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2023. Explainability for Large Language Models: A Survey. arXiv:2309.01029 [cs.CL] https://arxiv.org/abs/2309.01029