Mechanistic Interpretability for AI Safety
A Review

Anonymous authors
Paper under double-blind review

Abstract

As artificial intelligence (AI) systems rapidly advance, understanding their inner workings is crucial for ensuring alignment with human values and safety. This review explores mechanistic interpretability, which aims to reverse-engineer the computational mechanisms and representations learned by neural networks into human-understandable algorithms and concepts, focusing on a granular, causal understanding of how AI models operate. We establish foundational concepts, including features as units encoding knowledge within neural activations and hypotheses surrounding their representation and computation. We survey methodologies for causally dissecting model behaviors and assess the relevance of mechanistic interpretability to AI safety. We examine benefits in understanding, control, and alignment while discussing risks like capability gains and dual-use concerns. We examine the challenges of scalability, automation, and comprehensive understanding. We advocate for future work clarifying core concepts, setting rigorous standards, scaling up techniques to handle complex models and behaviors, and expanding the scope to domains like vision and reinforcement learning.

1 Introduction

As AI systems become increasingly sophisticated and general (Bubeck et al., 2023), advancing our understanding of these systems is crucial to ensure their alignment with human values and avoid catastrophic outcomes. The field of interpretability aims to demystify the internal processes of AI models, moving beyond evaluating performance alone. This review focuses on mechanistic interpretability, an emerging approach within the broader interpretability landscape that strives to specify the computations underlying deep neural networks comprehensively. We emphasize that understanding and interpreting these complex systems is not merely an academic endeavor; it's a societal imperative to ensure AI remains beneficial and trustworthy.

The interpretability landscape is undergoing a paradigm shift akin to the evolution from behaviorism to cognitive neuroscience in psychology. Historically, lacking tools for introspection, psychology treated the mind as a black box, focusing solely on observable behaviors. Similarly, interpretability has predominantly relied on black-box techniques, analyzing models based on input-output relationships or using attribution methods that, while probing deeper, still neglect the model's internal architecture. However, just as advancements in neuroscience allowed for a deeper understanding of internal cognitive processes, the field of interpretability is now moving towards a more granular approach. This shift from surface-level analysis to a focus on the internal mechanics of deep neural networks characterizes the transition towards inner interpretability (Räuker et al., 2023).

Mechanistic interpretability, as an approach to inner interpretability, aims to completely specify a neural network's computation, potentially in a format as explicit as pseudocode (also called *reverse-engineering*), striving for a granular and precise understanding of model behavior. It distinguishes itself primarily through its *ambition* for comprehensive reverse-engineering and its strong *motivation* towards AI safety. Our review serves as the first comprehensive exploration of mechanistic interpretability research, with the most accessible introductions currently scattered in a blog or list format (Olah, 2022; Nanda, 2022d; Olah et al., 2020; Sharkey et al., 2022a; Olah et al., 2018; Nanda, 2023f). We aim to synthesize the research (addressing the

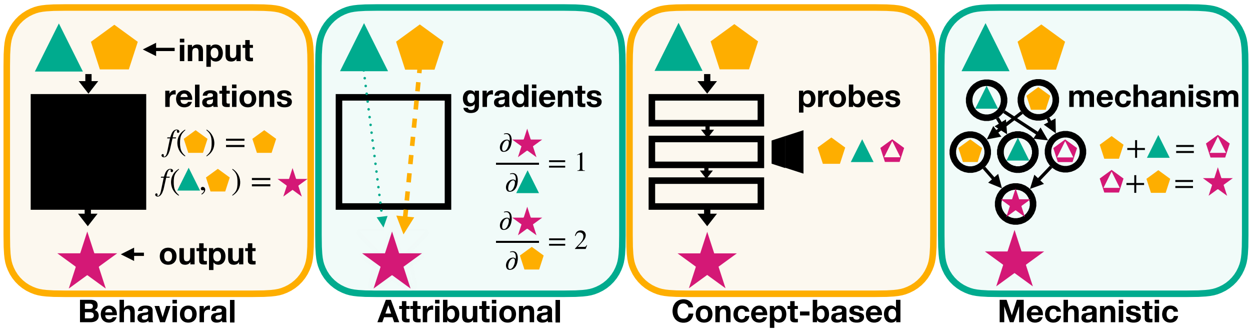


Figure 1: Interpretability paradigms offer distinct lenses for understanding neural networks: **Behavioral** analyzes input-output relations; **Attributional** quantifies individual input feature influences; **Concept-based** identifies high-level representations governing behavior; **Mechanistic** uncovers precise causal mechanisms from inputs to outputs.

"research debt" (Olah & Carter, 2017)) and provide a structured, accessible introduction for AI researchers and practitioners.

The structure of this paper provides a cohesive overview of mechanistic interpretability, situating the mechanistic approach in the broader interpretability landscape (Section 2), presenting core concepts and hypotheses (Section 3), explaining methods and techniques (Section 4), discussing evaluation (Section 5), presenting a taxonomy and survey of the current field (Section 6), exploring relevance to AI safety (Section 7), and addressing challenges (Section 8) and future directions (Section 9).

2 Interpretability Paradigms from the Outside In

We encounter a spectrum of interpretability paradigms for decoding AI systems' decision-making, ranging from external black-box techniques to internal analyses. We contrast these paradigms with mechanistic interpretability, highlighting its distinct causal bottom-up perspective within the broader interpretability landscape (see Figure 1).

Behavioral interpretability treats the model as a black box, analyzing input-output relations. Techniques such as minimal pair analysis (Warstadt et al., 2020), sensitivity and perturbation analysis (Casalicchio et al., 2018) examine input-output relations to assess the model's robustness and variable dependencies (Shapley, 1988; Ribeiro et al., 2016; Covert et al., 2021). Its *model-agnostic* nature is practical for complex or proprietary models but lacks insight into internal decision processes and causal depth (Jumelet, 2023).

Attributional interpretability aims to explain outputs by tracing predictions to individual input contributions using gradients. Raw gradients can be discontinuous or sensitive to slight perturbations. Therefore, techniques such as SmoothGrad (Smilkov et al., 2017) and Integrated Gradients (Sundararajan et al., 2017) average across gradients. Other popular techniques are layer-wise relevance propagation (Bach et al., 2015), DeepLIFT (Shrikumar et al., 2017), or GradCAM (Selvaraju et al., 2016). Attribution enhances transparency by showing input feature influence without requiring an understanding of the internal structure, enabling decision validation, compliance, and trust while serving as a bias detection tool.

Concept-based interpretability adopts a top-down approach to unraveling a model's decision-making processes by probing its learned representations for high-level concepts and patterns governing behavior. Techniques in this paradigm range from training supervised auxiliary classifiers (Belinkov, 2021) to employing unsupervised contrastive and structured probes (see Section 4.1) that explore the model's latent knowledge (Burns et al., 2023). Beyond observational analysis, concept-based interpretability can enable manipulation of these representations (*representation engineering* (Zou et al., 2023)), potentially enhancing safety by upregulating concepts such as honesty, harmlessness, and morality.

Mechanistic interpretability is a bottom-up approach that studies the fundamental components of models through granular analysis of features, neurons, layers, and connections, offering an intimate view of operational mechanics. Unlike concept-based interpretability, it aims to uncover causal relationships and precise computations transforming inputs into outputs, often identifying specific neural circuits driving behavior. This reverse-engineering approach draws from interdisciplinary fields like physics, neuroscience, and systems biology to guide the development of transparent, value-aligned AI systems. Mechanistic interpretability is the primary focus of this review.

3 Core Concepts and Assumptions

This section introduces the foundational concepts and hypotheses that underpin mechanistic interpretability, including the notion of features as fundamental units of representation, and their computation through circuits (Section 3.1), and the implications of these concepts for understanding the emergent properties of neural networks (Section 3.2).

3.1 Fundamental Units of Representation

Defining Features. The notion of a *feature* in neural networks is a central yet elusive concept, reflecting the pre-paradigmatic state of the field. Traditionally, features are understood as *characteristics or attributes of the input data stream* (Bishop, 2006). However, a broader interpretation suggests that a feature can be *any measurable property or characteristic of a phenomenon*, extending beyond human-interpretable elements.

The understanding of features encompasses two perspectives: human-centric and non-human-centric. A human-centric definition posits that *features are semantically meaningful, articulable properties of the input, encoded in activation space* (Olah, 2022). This view, however, may exclude features that are not understandable to humans. Adversarial examples have been interpreted as evidence for non-interpretable features that are perceptible to neural networks but not to humans (Ilyas et al., 2019). Furthermore, as neural networks evolve to surpass human capabilities, there is no inherent constraint that the features they learn must be comprehensible by humans; instead, they might discover increasingly abstract and alien features (Hubinger, 2019a).

A non-human-centric perspective defines *features as independent yet repeatable units that a neural network representation can decompose into* (Olah, 2022). This perspective allows for a more comprehensive understanding, encompassing features that are not necessarily interpretable by humans.

We adopt the notion of features as the smallest units of how neural networks encode knowledge, such that features cannot be further decomposed into smaller, distinct concepts. These features hypothetically serve as core components of a neural network's representation, analogous to how cells form the fundamental unit of biological organisms (Olah et al., 2020).

Definition: Feature

Features are the fundamental units of neural network representations.

Neurons as Computational Units? In the architecture of neural networks, *neurons* are the natural computational units, potentially representing individual features. Within a neural network representation $h \in \mathbb{R}^n$, the n basis directions are called neurons. For a neuron to be meaningful, the basis directions must functionally differ from other directions in the representation, forming a *privileged basis* – where the basis vectors are architecturally distinguished within the neural network layer from arbitrary directions in activation space. Typical non-linear activation functions privilege the basis directions formed by the neurons, making it meaningful to analyze individual neurons (Elhage et al., 2022b). Analyzing neurons can give insights into a network's functionality (Sajjad et al., 2022; Mu & Andreas, 2020; Dai et al., 2022; Ghorbani & Zou, 2020; Voita et al., 2023; Durrani et al., 2020; Goh et al., 2021; Bills et al., 2023; Huang et al., 2023).

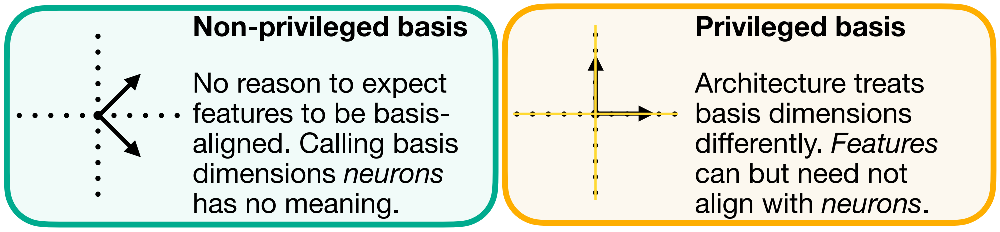

Non-privileged basis
No reason to expect features to be basis-aligned. Calling basis dimensions *neurons* has no meaning.
Privileged basis
Architecture treats basis dimensions differently. *Features* can but need not align with *neurons*.

Figure 2: Comparison of privileged and non-privileged basis in neural networks. Figure adapted from (Bricken et al., 2023).

Monosemantic and Polysemantic Neurons. A neuron corresponding to a single semantic concept is called *monosemantic*. The intuition behind this term comes from analyzing what inputs activate a given neuron, revealing its associated semantic meaning or concept. If neurons were the fundamental primitives of neural network representations, all neurons would be monosemantic, implying a one-to-one relationship between neurons and features. Comprehensive interpretability would be as tractable as characterizing all neurons and their connections. However, empirically, especially for transformer models (Elhage et al., 2022b), neurons are often observed to be *polysemantic*, *i.e.*, associated with multiple, unrelated concepts (Arora et al., 2018; Mu & Andreas, 2020; Elhage et al., 2022a; Olah et al., 2020). For example, a single neuron may be activated by both images of cats and images of cars, suggesting it encodes multiple unrelated concepts. Polysemanticity contradicts the interpretation of neurons as fundamental primitives and, in practice, makes it challenging to understand the information processing of neural networks.

Exploring Polysemanticity: Hypotheses and Implications. To understand the widespread occurrence of polysemanticity in neural networks, several hypotheses have been proposed:

- One trivial scenario would be that feature directions are orthogonal but not aligned with the basis directions (neurons). There is no inherent reason to assume that features would align with neurons in a non-privileged basis, where the basis vectors are not architecturally distinguished. However, even in a privileged basis formed by the neurons, the network could represent features not in the standard basis but as linear combinations of neurons (see Figure 2).
- An alternative hypothesis posits that *redundancy due to noise* introduced during training, such as random dropout (Srivastava et al., 2014), can lead to redundant representations and, consequently, to polysemantic neurons (Marshall & Kirchner, 2024). This process involves distributing a single feature across several neurons rather than isolating it into individual ones, thereby encouraging polysemanticity.
- Finally, the *superposition hypothesis* addresses the limitations in the network's representative capacity — the number of neurons versus the number of crucial concepts. This hypothesis argues that the limited number of neurons compared to the vast array of important concepts necessitates a form of compression. As a result, an n-dimensional representation may encode features not with the n basis directions (neurons) but with the $\propto \exp(n)$ possible almost orthogonal directions (Elhage et al., 2022b), leading to polysemanticity.

Hypothesis: Superposition

Neural networks represent more features than they have neurons by encoding features in overlapping combinations of neurons.

Superposition Hypothesis. The superposition hypothesis suggests that neural networks can leverage high-dimensional spaces to represent more features than the actual count of neurons by encoding features in almost orthogonal directions. Non-orthogonality means that features interfere with one another. However, the benefit of representing many more features than neurons may outweigh the interference cost, mainly when concepts are sparse and non-linear activation functions can error-correct noise (Elhage et al., 2022b).

Toy Model of Superposition

A toy model (Elhage et al., 2022b) investigates the hypothesis that neural networks can represent more features than the number of neurons by encoding real-world concepts in a compressed manner. The model considers a high-dimensional vector $\mathbf{x}$, where each element x_i corresponds to a feature capturing a real-world concept, represented as a random vector with varying importance determined by a weight a_i. These features are assumed to have the following properties: 1) **Concept Sparsity**: Real-world concepts occur sparsely. 2) **More Concepts Than Neurons**: The number of potential concepts vastly exceeds the available neurons. 3) **Varying Concept Importance**: Some concepts are more important than others for the task at hand.
The input vector $\mathbf{x}$ represents features capturing these concepts, defined by a sparsity level S and an importance level a_i for each feature x_i, reflecting the sparsity and varying importance of the underlying concepts. The model dynamics involve transforming $\mathbf{x}$ into a hidden representation $\mathbf{h}$ of lower dimension, and then reconstructing it as $\mathbf{x}'$:

$$\mathbf{h} = W\mathbf{x}, \quad \mathbf{x}' = \mathrm{ReLU}(W^T\mathbf{h} + \mathbf{b})$$

The network's performance is evaluated using a loss function $\mathcal{L}$ weighted by the feature importances a_i, reflecting the importance of the underlying concepts:

$$\mathcal{L} = \sum_x \sum_i a_i (x_i - x_i')^2$$

This toy model highlights neural networks' ability to encode numerous features representing real-world concepts into a compressed representation, providing insights into the superposition phenomenon observed in neural networks trained on real data.

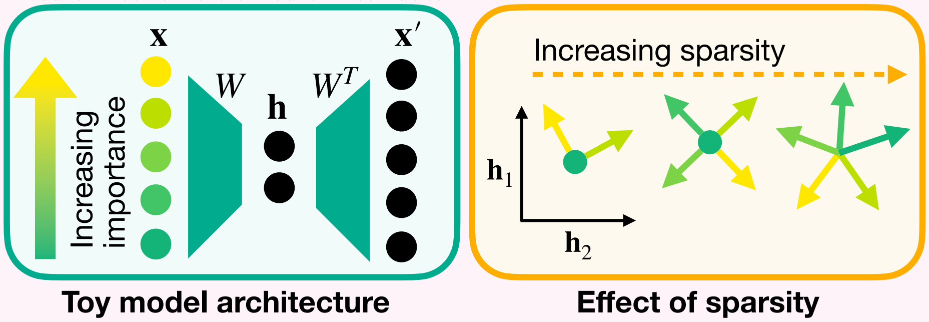


Figure 3: Illustration of the toy model architecture and the effects of sparsity. (left) Transformation of a five-feature input vector $\mathbf{x}$ into a two-dimensional hidden representation $\mathbf{h}$, and its reconstruction as $\mathbf{x}'$ using the weight matrix W and its transpose, with feature importance indicated by a color gradient from yellow to green. (right) The effect of increasing feature sparsity S on the encoding capacity of the network, highlighting the network's enhanced ability to represent features in superposition as sparsity increases from 0 to 0.9, illustrated by arrows in the activation space $\mathbf{h}$, which correspond to the columns of the matrix W.

Toy models can demonstrate under which conditions superposition occurs (Elhage et al., 2022b; Scherlis et al., 2023). Neural networks, via superposition, may effectively simulate computation with more neurons than they possess by allocating each feature to a linear combination of neurons, creating what is known as an overcomplete linear basis in the representation space. This perspective on superposition suggests that polysemantic models could be seen as compressed versions of hypothetically larger neural networks where each neuron represents a single concept (see Figure 4). Consequently, an alternative definition of features emerges:

Definition Feature (Alternative)

Features are elements that a network would ideally assign to individual neurons if neuron count were not a limiting factor (Bricken et al., 2023). In other words, features correspond to the disentangled concepts that a larger, sparser network with sufficient capacity would learn to represent with individual neurons.

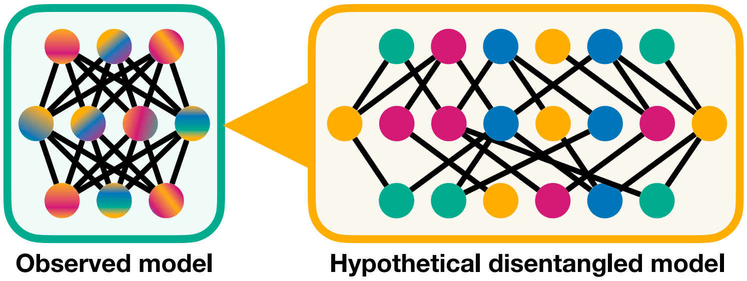


Figure 4: Observed neural networks (left) can be viewed as compressed simulations of larger, sparser networks (right) where neurons represent distinct features. An "almost orthogonal" projection compresses the high-dimensional sparse representation, manifesting as polysemantic neurons involved with multiple features in the lower-dimensional observed model, reflecting the compressed encoding. Figure adapted from (Bricken et al., 2023).

Research on superposition, including works by (Elhage et al., 2022b; Scherlis et al., 2023; Henighan et al., 2023), often investigates simplified models. However, understanding superposition in practical, transformer-based scenarios is crucial for real-world applications, as pioneered by (Gurnee et al., 2023).

The need for understanding networks despite polysemanticity has led to various approaches: One involves training models without superposition (Jermyn et al., 2022), for example, using a softmax linear unit (Elhage et al., 2022a) as an activation function to empirically increase the number of monosemantic neurons, but at the cost of making other neurons less interpretable. From a capabilities standpoint, polysemanticity may be desirable as it allows models to represent more concepts with limited compute, making training cheaper. Overall, engineering monosemanticity has proven challenging (Bricken et al., 2023) and may be impractical until we have orders of magnitude more compute available.

Another approach is to train networks in a standard way (creating polysemanticity) and use post-hoc analysis to find the feature directions in activation space, for example, with Sparse Autoencoders (SAEs). SAEs aim to find the true, disentangled features in an uncompressed representation by learning a sparse overcomplete basis that describes the activation space of the trained model (Bricken et al., 2023; Sharkey et al., 2022b; Cunningham et al., 2024) (also see Section 4.1).

If not neurons, what are features then? We want to identify the fundamental units of neural networks, which we call *features*. Initially, neurons seemed likely candidates. However, this view fell short, particularly in transformer models where neurons often represent multiple concepts, a phenomenon known as polysemanticity. The superposition hypothesis addresses this, proposing that due to limited representational capacity, neural networks compress numerous features into the confined space of neurons, complicating interpretation.

This raises the question: *How are features encoded if not in discrete neuron units?* While a priori features could be encoded in an arbitrarily complex, non-linear structure, a growing body of theoretical arguments and empirical evidence supports the hypothesis that features are commonly represented linearly, i.e., as linear combinations of neurons - hence, as directions in representation space. This perspective promises to enhance our comprehension of neural networks by providing a more interpretable and manipulable framework for their internal representations.

Hypothesis: Linear Representation

Features are directions in activation space, *i.e.* as linear combinations of neurons.

The linear representation hypothesis is central to neural network analysis, suggesting that networks represent high-level features as linear directions in activation space. This hypothesis simplifies the neural network representations, enhancing their interpretability and ease of manipulation (Nanda et al., 2023b).

The architecture of neural networks, typically comprising linear layers interspersed with non-linear activation functions, inherently favors linear representations. When a neural network layer processes the information from previous layer activations, it typically employs matrix multiplication - only linear features can be processed in a subsequent single linear layer. Conversely, more complex non-linear encodings, though theoretically possible, would require multiple layers to be decoded. Hence, even hypothetical non-linear representations are reducible to intermediate linear representations in the typical neural network design.

A notable case study involving the GPT model's application to Othello game prediction suggested a non-linear internal representation of the board state, as decoding required a two-layer probe (Li et al., 2023a). However, further analysis revealed it to be a linear representation. Interestingly, the linear representation was not of the "black" and "white" pieces, as probed initially, but instead of "my own" and "the opponent's" pieces, which, in hindsight, were more relevant features for the model to represent (Nanda et al., 2023b). To date, there is no evidence of non-linear representations in neural networks.

Empirical evidence supports the linear representation hypothesis: Firstly, the seminal work by Mikolov et al. (2013) revealing semantic vector calculus in word embeddings pointed to linear representations. Interpretability methods like linear probing (Alain & Bengio, 2016; Belinkov, 2021) and sparse dictionary learning (Bricken et al., 2023; Cunningham et al., 2024; Deng et al., 2023) confirm the linear accessibility of meaningful features. It is possible to decode concepts (O'Mahony et al., 2023), tasks (Hendel et al., 2023), functions (Todd et al., 2023), sentiment (Tigges et al., 2023), and relations (Hernandez et al., 2023; Chanin et al., 2023) linearly in large language models. Additionally, breakthroughs in linear addition for model steering (Turner et al., 2023; Sakarvadia et al., 2023a; Li et al., 2023b) and representation engineering (Zou et al., 2023) highlight the practical implications of linear feature representations regarding model manipulation and interpretability.

While the linear representation hypothesis facilitates interpretability significantly, Sharkey et al. (2022a) warns against neglecting the potential role of non-linear representations. Given the dynamic nature of neural network development, it's crucial to continuously reevaluate the hypothesis, particularly in light of the possible emergence of non-linear features when interpretability tools that rely on linear representations are subject to optimization pressure (Hubinger, 2022). In this context, the polytope lens provides an alternative perspective, as proposed by Black et al. (2022). This approach shifts the focus to the impact of non-linear activation functions, examining how discrete polytopes, formed by piecewise linear activations, might be the fundamental primitives of neural network representation.

3.2 Computation and Abstraction

Having defined features as directions in activation space as the fundamental units of neural network representation, we now explore their computation. Neural networks can be conceptualized as computational graphs, within which *circuits* are sub-graphs consisting of linked features and the weights connecting them. Similar to how features are the representational primitive, circuits function as the computational primitive (Michaud et al., 2023) and the primary building block of these networks (Olah et al., 2020).

The decomposition of neural networks into circuits for interpretability has shown significant promise, particularly in small models trained for specific tasks such as addition, as seen in the work of Nanda et al. (2023a) and Quirke & Barez (2023). However, scaling this analysis to broader behaviors remains challenging. To

Definition: Circuit

Circuits are sub-graphs of the network, consisting of linked features and the weights connecting them.

date, only relatively narrow behaviors like Python docstring formatting (Heimersheim & Jett, 2023) and greater-than-computations (Hanna et al., 2023) have been thoroughly analyzed.

Despite these challenges, there has been notable progress in scaling circuit analysis to larger circuits, such as on GPT-2's indirect object identification (Wang et al., 2023) and on scaling to larger models such as multiple-choice question answering in Chinchilla (Lieberum et al., 2023). The circuits underlying more general and transferable behaviors are also being explored: McDougall et al. (2023)'s research on copy suppression in GPT-2's attention heads, for instance, sheds light on model calibration and self-repair mechanisms. Similarly, Davies et al. (2023) and Feng & Steinhardt (2023) focus on how LLMs perform variable binding and entity-attribute binding, respectively, providing insights into the representation of symbolic knowledge. Yu et al. (2023) explore mechanisms for factual recall in LLMs, revealing how circuits dynamically balance pre-trained knowledge with new contextual information. Lan & Barez (2023) extend circuit analysis to sequence continuation tasks, identifying shared computational structures across semantically related sequences, thereby enriching our understanding of error identification.

More promisingly, some repeating patterns have shown universality across models and tasks. These universal patterns are called motifs (Olah et al., 2020) and can manifest not just as specific circuits or features but also as higher-level behaviors emerging from the interaction of multiple components. Examples include the curve detectors found across vision models (Cammarata et al., 2021; 2020), induction circuits enabling in-context learning (Olsson et al., 2022), and the phenomenon of branch specialization in neural networks (Voss et al., 2021). Motifs may also capture how models leverage tokens for working memory or parallelize computations in a divide-and-conquer fashion across representations. The significance of motifs lies in revealing the common structures, mechanisms, and strategies that naturally emerge across neural architectures, shedding light on the fundamental building blocks underlying their intelligence.

Definition: Motif

Motifs are repeated patterns within a network, encompassing either features or circuits that emerge across different models and tasks.

Universality Hypothesis. Following the evidence for motifs or repeated patterns in neural networks, the universality hypothesis emerges as a pivotal concept. This hypothesis posits a convergence in forming features and circuits across various models and tasks, which could significantly ease interpretability efforts in AI. The universality hypothesis proposes that artificial and biological neural networks share similar features and circuits, suggesting a standard underlying structure (Chan et al., 2023; Sucholutsky et al., 2023; Kornblith et al., 2019). This idea posits that there is a fundamental basis in how neural networks, irrespective of their specific configurations, process and comprehend information. This could be due to inbuilt inductive biases in neural networks or *natural abstractions* (Chan et al., 2023) – concepts favored by the natural world that any cognitive system would naturally gravitate towards.

Hypothesis: Universality

Analogous features and circuits form across models and tasks (Olah et al., 2020).

Evidence for this hypothesis comes from cross-species neural structures in neuroscience, where similar neural structures and functions are found in different species (Kirchner, 2023). Additionally, machine learning models, including neural networks, tend to converge on similar features, representations, and classifications across different tasks and architectures (Chen et al., 2023a; Hacohen et al., 2020; Li et al., 2015; Bricken et al., 2023). While various studies support the universality hypothesis, questions remain about the extent of feature and circuit similarity across different models and tasks. Nevertheless, this concept bridges AI and other scientific disciplines, offering cross-disciplinary applications and a deeper understanding of artificial and natural cognitive processes. In the context of mechanistic interpretability, this hypothesis has been investigated for neurons (Gurnee et al., 2024), group composition circuits (Chughtai et al., 2023), and modular task processing (Variengien & Winsor, 2023).

Internal World Models. World models are internal causal models of an environment formed within neural networks. Traditionally linked with reinforcement learning, these models are *explicitly* trained to develop a compressed spatial and temporal representation of the training environment, enhancing downstream task performance and sample efficiency through training on internal hallucinations (Ha & Schmidhuber, 2018). However, in the context of our survey, our focus shifts to world models that potentially form *implicitly* as a by-product of the training process, especially in LLMs that are trained on next-token prediction - also called generative pre-trained transformers (GPT).

A critical perspective often surfacing in discussions about LLMs is their characterization as *stochastic parrots* (Bender et al., 2021). This label stems from their fundamental operational mechanism of predicting the next word in a sequence, supposedly relying heavily on memorization. From this viewpoint, LLMs are seen as forming complex correlations based on observational data but are thought to lack the ability to develop causal models of the world. This limitation is attributed to their lack of access to interventional data (Pearl, 2009).

However, this understanding of LLMs shifts significantly when viewed through the lens of the active inference framework (Salvatori et al., 2023), a theory rooted in cognitive science and neuroscience. Active inference postulates that the objective of minimizing prediction error, given enough representative capacity, is adequate for a learning system to develop complex world representations, behaviors, and abstractions. Since language inherently mirrors the world, these models could implicitly construct linguistic and broader world models. This perspective presupposes that LLMs, in their pursuit of better modeling sequences, inherently learn world models, abstractions, and algorithms for this purpose (Kulveit et al., 2023).

This alternative understanding of LLMs aligns with the simulation hypothesis, which suggests that models designed for prediction, such as LLMs, will eventually simulate the causal processes underlying data creation. Seen as an extension of their drive for efficient compression, this hypothesis implies that adequately trained models like GPT could develop internal world models as a natural outcome of their predictive training (janus, 2022).

Hypothesis: Simulation

A model whose objective is text prediction will simulate the causal processes underlying the text creation if optimized sufficiently strongly (janus, 2022).

In addition to theoretical considerations for emergent causal world models (Richens & Everitt, 2024; Nichani et al., 2024), mechanistic interpretability is starting to provide empirical evidence on the types of internal world models that may emerge in LLMs. The ability to internally represent the board state in games like Othello (Li et al., 2023a; Nanda et al., 2023b), create linear abstractions of spatial and temporal data (Gurnee & Tegmark, 2023), and structure complex representations of mazes, demonstrating an understanding of maze topology and pathways (Ivanitskiy et al., 2023) highlight the growing abstraction capabilities of LLMs.

These emergent world models have significant implications for AI alignment research. For example, finding an internal representation of human values and aiming the AI systems objective may be the most trivial way to achieve alignment (Wentworth, 2022). Especially if the world model is internally separated from notions of goals and agency (Ruthenis, 2022), world model interpretability may be enough for alignment (Ruthenis, 2023).

Conditioning of pre-trained models as a pathway towards general intelligence is deemed comparatively safe, as it avoids directly creating agents with inherent goals or agendas (Jozdien, 2022; Hubinger et al., 2023).

However, Hubinger et al. (2023) also highlights that prompting a model to simulate an actual agent, such as "You are a superintelligence in 2035 writing down an alignment solution:", could inadvertently lead to the formation of internal agents. In contrast, training with reinforcement learning tends to create agents by default (Casper et al., 2023a; Ngo et al., 2022).

The prediction orthogonality hypothesis further expands on this idea: It posits that prediction-focused models like GPT may simulate agents with various objectives and levels of optimality. In this context, GPT is a simulator, simulating entities known as simulacra that can be either agentic or non-agentic, with different objectives from the simulator itself (janus, 2022).

Hypothesis: Prediction Orthogonality

A model whose objective is prediction can simulate agents who optimize toward any objectives with any degree of optimality (janus, 2022).

This prediction orthogonality hypothesis suggests that models primarily focused on prediction, such as GPT, can simulate agents — referred to as 'simulacra' — with potentially misaligned objectives (janus, 2022). Although GPT itself lacks genuine agency or intentionality, it may produce outputs that simulate these qualities (Bereska & Gavves, 2023), underscoring the need for careful oversight and, better yet, finding internal agents or their constituents such as optimization or search potentially via mechanistic interpretability - an endeavor also known as searching for search (NicholasKees & janus, 2022).

In conclusion, the evolution of LLMs from simple predictive models to entities potentially possessing complex internal world models, as suggested by the simulation hypothesis and supported by mechanistic interpretability studies, represents a significant shift in our understanding of these systems. This evolution challenges us to reconsider LLMs' capabilities and future trajectories in the broader landscape of AI development.

4 Core Methods

Mechanistic interpretability employs tools and techniques adopted from various interpretability approaches, focusing on causal methods that distinguish it from traditional, more observational techniques. This section provides an overview of the essential methodologies, enabling detailed observation and analysis of neural network models (Section 4.1), as well as interventional methods that allow for direct manipulation within the model (Section 4.2). The interplay between observation and intervention facilitates a comprehensive understanding of neural network operations (Section 4.3). Figure 5 provides an overview of the relevant methods and techniques.

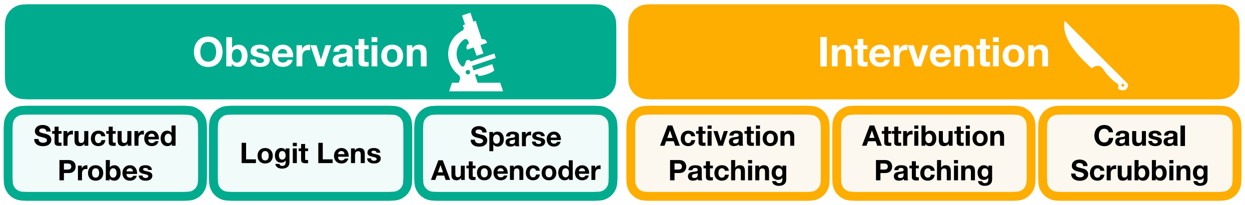


Figure 5: Overview of relevant methods and techniques employed in mechanistic interpretability research. Observational methods proposed for mechanistic interpretability include structured probes (more aligned with top-down interpretability), logit lens variants, and sparse autoencoders (SAEs). Additionally, as mechanistic interpretability focuses on causal understanding, novel methods encompass variants of activation patching for uncovering causal mechanisms and causal scrubbing for hypothesis evaluation.

4.1 Observation

Mechanistic interpretability draws from observational methods that analyze the inner workings of neural networks, with many of these methods preceding the field itself. For a detailed exploration of inner interpretability methods, refer to (Räuker et al., 2023). Two prominent categories are example-based methods and feature-based methods:

- **Example-based methods** identify real input examples that highly activate specific neurons or layers. This helps pinpoint influential data points that maximize neuron activation within the neural network.
- **Feature-based methods** encompass techniques that generate synthetic inputs to optimize neuron activation. These neuron visualization techniques reveal how neurons respond to stimuli and which features are sensitive to (Zeiler & Fergus, 2014). By understanding the synthetic inputs that drive neuron behavior, we can hypothesize about the features encoded by those neurons.

Probing for Features Probing involves training a classifier using the activations of a model, with the classifier's performance subsequently observed to deduce insights about the model's behavior and internal representations. As highlighted by Belinkov (2021), this technique faces a notable challenge: the probe's performance may often reflect its own learning capacities more than the actual characteristics of the model's representations. This dilemma has led researchers to investigate the ideal balance between the complexity of a probe and its capacity to accurately represent the model's features (Cao et al., 2021; Voita & Titov, 2020).

The linear representation hypothesis offers a resolution to this issue. Under this hypothesis, the failure of a simple linear probe to detect certain features suggests their absence in the model's representations. Conversely, if a more complex probe succeeds where a simpler one fails, it implies that the model contains features that a complex function can combine into the target feature. Still, the target feature itself is not explicitly represented. This hypothesis implies that using linear probes could suffice in most cases, circumventing the complexity considerations generally associated with probing (Belinkov, 2021).

An illustrative example can be seen in the work of Li et al. (2023a), who demonstrated that the internal representation of a GPT model trained on the Othello board game could be decoded only with a non-linear probe, not a linear one. This finding suggests that the explicit representation of the board state in terms of "black" and "white" pieces is not present linearly, but other features implicitly represent it. Complementing this, Nanda (2023c) showed that when decoding the board state in terms of "own" and "opponent's" pieces, a linear probe suffices, thereby reaffirming the linear representation hypothesis under certain conditions.

A significant limitation of probing is the inability to draw behavioral or causal conclusions. The evidence provided by probing is mainly observational, focusing on what information is encoded rather than how it is used (also see Figure 1). This necessitates careful analysis and possibly the adoption of alternative approaches (Elazar et al., 2021) or the integration of intervention techniques to draw more substantive conclusions about the model's behavior (Section 4.2).

Probing has been used to analyze the acquisition of chess knowledge in AlphaZero (McGrath et al., 2022) and the representation of linguistic information in BERT (Tenney et al., 2019). Gurnee et al. (2023) introduce *sparse probing*, decoding internal neuron activations in large models to understand feature representation and sparsity. They show that early layers use sparse combinations of neurons to represent many features in superposition, while middle layers have dedicated monosemantic neurons for higher-level contextual features.

Structured Probes While most of this review focuses on bottom-up, mechanistic approaches to interpretability, it is worth considering the potential for integrating top-down, concept-based techniques like structured probes. Structured probes represent an advanced technique in conceptual interpretability, playing a crucial role in probing language models to uncover complex features like truth representations.

A notable advancement in this domain is the discovery of an "internal truth" direction within language models using unsupervised contrastive probing, as demonstrated by the contrast-consistent search (CCS) method proposed by Burns et al. (2023). CCS identifies linear projections of hidden states that exhibit logical consistency, ensuring contrasting truth values for statements and their negations, contributing substantially to conceptual interpretability (Zou et al., 2023).

However, structured probes face significant challenges, particularly in unsupervised probing scenarios. A major concern is verifying the accuracy of discovered features, as unsupervised methods can identify numerous features without a straightforward verification process. Additionally, recent work by Farquhar et al. (2023) raises doubts about the scalability of CCS, suggesting that the CCS loss may capture simulated knowledge rather than the model's true knowledge, especially in highly capable models adept at simulating agents (simulacra) with their own belief systems.

While structured probes primarily focus on high-level conceptual representations, their findings could potentially inform or complement mechanistic interpretability efforts. For instance, identifying truth directions through structured probes could help guide targeted interventions or analyze the underlying circuits responsible for truthful behavior using mechanistic techniques like activation patching or circuit tracing (Section 4.2). Conversely, mechanistic methods could provide insights into how truth representations emerge and are computed within the model, addressing some of the challenges faced by unsupervised structured probes.

Logit Lens The *logit lens* (nostalgebraist, 2020) provides a window into the model's predictive process by applying the final layer's linear function to intermediate layers, revealing how prediction confidence evolves across computational stages. Extensions of this approach include the tuned lens (Belrose et al., 2023), which trains affine probes to decode hidden states into probability distributions over the vocabulary, and the Future Lens (Pal et al., 2023), which explores the extent to which individual hidden states encode information about subsequent tokens.

Researchers have also investigated techniques that bypass intermediate computations to probe representations directly. Din et al. (2023) propose using linear transformations to approximate hidden states from different layers, revealing that language models often predict final outputs in early layers. Dar et al. (2022) present a theoretical framework for interpreting transformer parameters by projecting them into the embedding space, enabling model alignment and parameter transfer across architectures.

Other techniques focus on interpreting specific model components or submodules. The DecoderLens (Langedijk et al., 2023) allows analyzing encoder-decoder transformers by cross-attending intermediate encoder representations in the decoder, shedding light on the information flow within the encoder. The Attention Lens (Sakarvadia et al., 2023b) aims to elucidate the specialized roles of attention heads by translating their outputs into vocabulary tokens via learned transformations.

Feature Disentanglement via Sparse Dictionary Learning As highlighted in Section 3.1, recent work suggests that the essential elements in neural networks are linear combinations of neurons representing features in superposition (Elhage et al., 2022b). Sparse autoencoders provide a methodology to decompose neural network activations into these individual component features (Sharkey et al., 2022b; Cunningham et al., 2024). This process involves reconstructing activation vectors as sparse linear combinations of directional vectors within the activation space, a problem also known as sparse dictionary learning (Olshausen & Field, 1997).

Sparse dictionary learning has led to the development of various sparse coding algorithms (Lee et al., 2006). The sparse autoencoder stands out for its simplicity and scalability (Sharkey et al., 2022b). The first application to a language model was by Yun et al. (2021), who implemented sparse dictionary learning across multiple layers of a language model.

Sparse autoencoders, a variant of the standard autoencoder framework, incorporate sparsity regularization to encourage learning sparse yet meaningful data representations. Theoretical foundations in the disentanglement literature suggest that autoencoders can recover ground truth features under feature sparsity and non-negativity (Whittington et al., 2022).

Practical implementations, such as the toy model by Sharkey et al. (2022b), demonstrate the viability of this approach, with the precise tuning of the sparsity penalty on the hidden activations being a critical aspect that dictates the sparsity level of the autoencoder (Sharkey et al., 2022b). We show an overview in the pink box on sparse autoencoders in Figure 6.

Sparse Dictionary Learning

Sparse autoencoders (Cunningham et al., 2024) are a solution to the sparse dictionary learning (Olshausen & Field, 1997) problem to decompose neural network activations into individual component features. The goal is to learn a dictionary of vectors $\{\mathbf{f}_k\}_{k=1}^{n_{\text{feat}}} \subset \mathbb{R}^d$ that can represent the unknown, ground truth network features $\{\mathbf{g}_j\}_{j=1}^{n_{\text{gt}}}$ as sparse linear combinations. The autoencoder architecture consists of an encoder and a ReLU activation function, expanding the input dimensionality to $d_{\text{hid}} = Rd_{\text{in}}$, where R controls the ratio of the feature dictionary size to the model dimension. The encoder's output is given by:

$$\mathbf{h} = \text{ReLU}(W_{\text{enc}}\mathbf{x} + \mathbf{b}) \tag{1}$$

$$\mathbf{x}' = W_{\text{dec}}\mathbf{h} = \sum_{i=0}^{d_{\text{hid}}-1} h_i \mathbf{f} i \tag{2}$$

where $W_{\text{enc}}, W_{\text{dec}}^T \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}$ and $\mathbf{b} \in \mathbb{R}^{d_{\text{hid}}}$. The parameter matrix W_{dec} forms the feature dictionary, with rows $\mathbf{f}_i$ as dictionary features. The autoencoder is trained to minimize the loss, where the L_1 penalty on $\mathbf{h}$ encourages sparse reconstructions using the dictionary features.

$$\mathcal{L}(\mathbf{x}) = ||\mathbf{x} - \mathbf{x}'||_2^2 + \alpha||\mathbf{h}||_1 \tag{3}$$

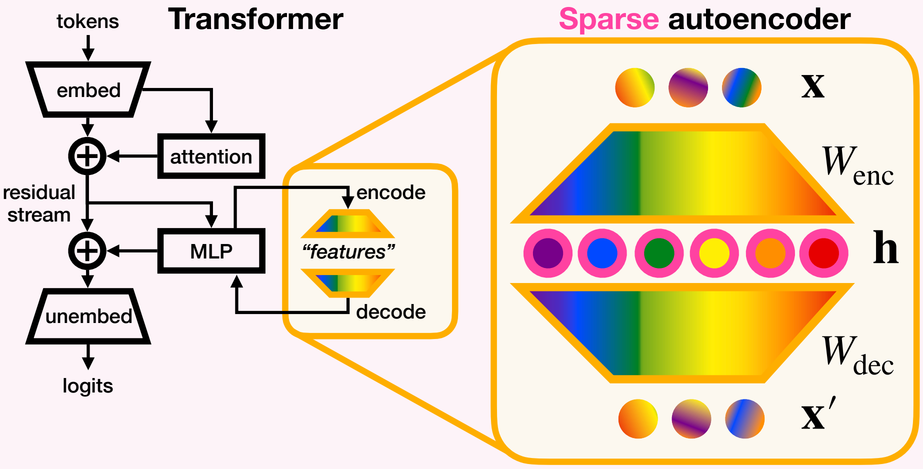


Figure 6: Illustration of a sparse autoencoder applied to the MLP layer activations, consisting of an encoder that increases dimensionality while emphasizing sparse representations and a decoder that reconstructs the original activations using the learned feature dictionary.

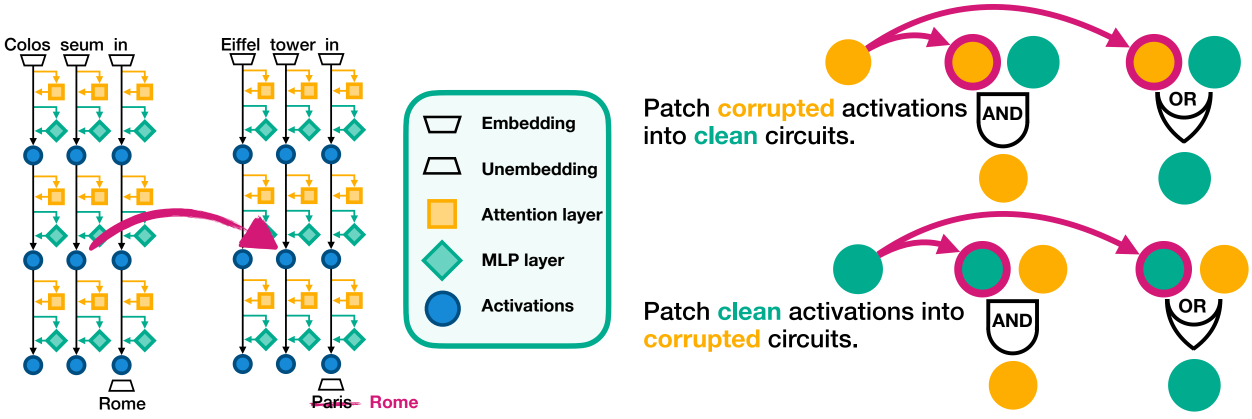
Empirical studies indicate that sparse autoencoders can enhance the interpretability of neural networks, exhibiting higher scores on the autointerpretability metric and increased monosemanticity (Bricken et al., 2023; Cunningham et al., 2024; Sharkey et al., 2022b). Furthermore, sparse autoencoders have been employed to measure feature sparsity (Deng et al., 2023) and interpret reward models in reinforcement learning-based language models (Marks et al., 2023), making them an actively researched area in mechanistic interpretability.

4.2 Intervention

Activation Patching is a collective term for a set of causal intervention techniques, also known as causal tracing (Meng et al., 2022a), interchange intervention (Geiger et al., 2021b), causal mediation analysis (Vig et al., 2020), and causal ablation (Wang et al., 2023). While nuanced in their application, these techniques share the common goal of manipulating neural network activations to shed light on the decision-making processes within the model.

Activation patching modifies a neural model's internal state by replacing specific activations with alternative values, such as zeros, mean activations across samples, random noise, or activations from a different forward pass. This technique enables researchers to isolate and examine the effects of modifying particular neural circuits within the model to understand how these circuits interact and contribute to the model's behavior in response to different inputs. By selectively replacing activations, activation patching highlights the circuits directly responsible for particular behaviors or outputs while reducing the influence of other, irrelevant circuits.

The primary goal is to isolate and understand the role of specific components or circuits within the model. By observing how changes in activations affect the model's output, researchers can infer the function and importance of those components. Critical applications are *(i)* localizing behavior by identifying critical activations, for example, understanding storage and processing of factual information (Meng et al., 2022a; Geva et al., 2023; Goldowsky-Dill et al., 2023; Stolfo et al., 2023), and *(ii)* analyzing component interactions, such as conducting circuit analysis to identify sub-networks within a model's computation graph that implement specified behaviors (Wang et al., 2023; Hanna et al., 2023; Lieberum et al., 2023; Hendel et al., 2023; Geva et al., 2023).



(a) The activation patching process

(b) Boolean logic analogies

Figure 7: (a) The transfer of activations from clean to corrupted inputs isolates neural circuits. (b) Boolean logic circuits are an analogy for sufficiency and necessity in neural circuits via patching strategies.

The standard protocol of activation patching entails: *i.)* running a model with a clean input and caching the latent activations; *ii.)* executing the model with a corrupted input; *iii.)* re-running the model with the corrupted input but substituting specific activations with those from the clean cache; and *iv.)* determining significance by observing the variations in the model's output during the third step, thereby highlighting the importance of the replaced components.

The process relies on comparing pairs of inputs: a *clean* input, which triggers the desired behavior, and a *corrupted* input, which is identical to the clean one except for critical differences that prevent the behavior. This careful selection ensures that the two inputs share as many circuits as possible, except those directly influencing the behavior under investigation - effectively *controlling* for confounding circuitry. Through activation patching—transferring activations from the clean input run to the corrupted one—researchers can maintain the shared circuits' functionality while pinpointing and isolating the specific circuit responsible for the behavior.

Differences in patching direction — clean to corrupted versus corrupted to clean — yield insights into which model components are *sufficient* or *necessary* for a given behavior. Clean to corrupted patching (causal tracing) identifies activations that are *sufficient* for restoring clean performance, highlighting the non-necessity of redundant components in achieving specific outputs. If many components redundantly encode something that quickly saturates, you can get good performance from patching in any of them, even if none are necessary. This approach, effective even in redundant system components, clarifies the sufficiency of specific activations in driving model performance under OR logic conditions: In circuits A-AND-B, this tells us nothing, but in A-OR-B, it tells us that both A or B is sufficient on its own.

Conversely, corrupted to clean patching (resample ablation) focuses on determining the *necessary* activations for clean performance, emphasizing the criticality of specific components. If the model has redundancy, we may see that nothing is necessary! Even if, in the aggregate, they're essential. Particularly useful in AND logic scenarios, this method assesses the impact of removing specific activations, revealing the indispensable elements of the computational architecture. In the circuit A-OR-B, resample ablating does nothing. However, A-AND-B tells us that removing each of A or B will dramatically reduce performance.

The approach employs various corruption methods, including zero-, mean-, random-, or resample ablation - replacing activations with zeros, an average over activations across diverse samples, Gaussian noise (Meng et al., 2022a), or activations from a different model run (Vig et al., 2020; Wang et al., 2023), each serving to modulate the model's internal state in distinct ways. Among these, resample ablation stands out for its effectiveness in maintaining consistent model behavior by not changing the data distribution too much (Zhang & Nanda, 2023). While breaking behavior is always possible by taking the model off-distribution, this is uninteresting for finding the relevant circuit (Nanda, 2023e). Therefore, one needs to be careful when interpreting the results of patching.

Among evaluation metrics for assessing how activation patching influences behavior, comparing the *logit difference* between clean and corrupted runs stands out as a precise measure of the changes in confidence levels across different inputs and the ability to detect negative modules (Zhang & Nanda, 2023). Additionally, *per-token log probability* provides a detailed view of the model's prediction confidence at each token, providing more granularity. *Direct logit attribution* further delves into how different components influence the logit of the correct next token, shedding light on the critical elements of the model's predictions. However, internal memory management can mislead this metric (Dao et al., 2023). When used together, these metrics enable a thorough evaluation of the impact of activation patching, offering comprehensive insights into the intervention's effects on model behavior.

Activation patching, while a powerful interpretability tool, encounters several limitations. A primary concern is its limited ability to generalize beyond specific distributions or tasks, often focusing on narrow scenarios without fully addressing broader or varied contexts (Nanda, 2023d). Another issue is the *MLP-In-The-Middle* illusion (Lange et al., 2023), a phenomenon where patching an entire Multi-Layer Perceptron (MLP) layer shows no observable effect, yet patching a specific subspace within the same layer reveals significant impacts. This raises questions about the relevance of certain subspaces in the model's normal functioning. This suggests that some components may appear crucial in patching but are dormant or irrelevant in regular operations.

Additionally, the *Hydra effect* (McGrath et al., 2023), where models internally self-repair and maintain capabilities even when key components are ablated, can sometimes obscure the relevant components. The effects of patching can propagate and interact in complex ways, potentially exaggerating or diminishing the apparent importance of certain components (see Section 8.2).

Translating the localization of model behaviors, as revealed by activation patching, into effective model editing (Hase et al., 2023) can also be challenging. Understanding where certain information or processes are stored in the model doesn't always seem to translate into actionable strategies for modifying or improving the model's performance or behavior.

Furthermore, the process of activation patching can be slow, which is especially problematic in large models or when attempting to automate the process (Conmy et al., 2023). However, this challenge can be partially mitigated using *attribution patching* (Nanda, 2023d; Syed et al., 2023), a gradient-based alternative that takes a linear approximation to traditional activation patching, similar to other attribution methods (see Section 2). Attribution patching offers a faster and more scalable approach, particularly advantageous in automated circuit discovery (Syed et al., 2023) and large model applications, providing a more feasible and efficient means of probing neural network behaviors while retaining the core benefits of the activation patching approach.

Recent advancements include the introduction of AtP* (Kramár et al., 2024), a refined version of attribution patching that addresses specific failure modes of the original method to reduce false negatives, thus improving its reliability while maintaining scalability. Other variations include path patching (Goldowsky-Dill et al., 2023), which quantitatively tests hypotheses expressing that behaviors are localized to a set of paths, and attention pattern patching (Nanda, 2023d), which leverages attention attribution patterns to gain insights into information flow within the network. Ghandeharioun et al. (2024) introduced a unified framework to analyze hidden representations.

4.3 Integrating Observation and Intervention

Integrating different methodologies is necessary for a thorough understanding of neural network models. These complex models require a broad approach that goes beyond individual techniques. An effective strategy combines feature-level analysis tools like sparse autoencoders with probing and interventional methods like activation patching. This integrated approach could allow a more in-depth examination of neural network feature-level circuits.

Even seemingly independent techniques can improve the robustness of analysis. For example, validating activation patching findings with maximally activating dataset examples or direct logit attribution offers a more comprehensive view of a component's network functionality. However, achieving complete understanding remains challenging due to the potential for feature superposition within these models. A single component may simultaneously represent multiple features, complicating interpretability efforts. Navigating and disentangling these intertwined representations requires integrating diverse analytical techniques.

5 Evaluation

Qualitative Evaluation. Interpretability research lacks established metrics, making qualitative results crucial. The *signal of structure* approach (Olah & Jermyn, 2024), observing intricate patterns indicating genuine structures, resembles examining cells under a microscope. A nuanced balance of qualitative observations and quantitative analyses is required, often necessitating custom interfaces to avoid oversimplification.

Quantitative Evaluation. A central challenge is the lack of rigorous evaluation methods. Relying solely on intuition is inadequate, as hypotheses can be conflated with conclusions (Rudin, 2019; Miller, 2019; Räuker et al., 2023), leading to cherry-picking and optimizing for best-case performance rather than aiming for methods that perform well on average or in worst-case scenarios (Casper, 2023) (see also Section 8.1). Current practices are ad hoc, with proxies (Doshi-Velez & Kim, 2017) potentially leading to over-optimization (Goodhart's law - "When a measure becomes a target, it ceases to be a good measure"). Distinguishing correlation from causation is crucial, as interpretability illusions (Bolukbasi et al., 2021; Olah et al., 2017) demonstrate visualizations may be meaningless without causal linking.

Rigorous evaluation methods are needed, such as *i.)* assessing out-of-distribution inputs, as most current methods are only valid for analyzing specific examples or datasets (Räuker et al., 2023; Ilyas et al., 2019; Mu & Andreas, 2020; Casper et al., 2023b; Burns et al., 2023), *ii.)* controlling the system through edits, such as implanting or removing trojans (Mazeika et al., 2022) or targeted editing (Ghorbani & Zou, 2020; Dai et al., 2022; Meng et al., 2022a;b; Bau et al., 2018; Hase et al., 2023), *iii.)* or replacing it with simpler reverse-engineered alternatives (Lindner et al., 2023). Ultimately, establishing benchmarks, ideally automated, is required.

Causality as a Theoretical Foundation. The theory of causality (Pearl, 2009) provides a mathematically precise language for mechanistic interpretability, forming the foundation for understanding high-level semantics in neural representations (Geiger et al., 2023a). Treating neural networks as causal models involves considering the compute graph as the causal graph, allowing for precise interventions and examining individual parameters' roles (McGrath et al., 2023). In contrast to typical real-world causal analyses, the causal model is known with complete certainty, long chains of interventions are possible, and all variable values can be simultaneously read. However, the large number of parameters, often lacking clear meaning, poses a challenge in this context.

Causal inference techniques have been employed in various contexts within neural networks, including locating factual knowledge (Meng et al., 2022a), addressing gender bias through mediation analyses (Vig et al., 2020), and constructing causal abstractions of neural network computations (Geiger et al., 2023a; 2021a; 2023b; 2021b; McGrath et al., 2023). Ablations and interchange interventions have been suggested as means to validate hypotheses about mechanisms in neural networks and enforce specific structures (Chan et al., 2022; Leavitt & Morcos, 2020; Geiger et al., 2021b), enabling large-scale analysis of model behavior (Wu et al., 2023).

Rigorous Hypothesis Testing. Causal scrubbing (Chan et al., 2022), causal abstraction (Geiger et al., 2023a), and locally consistent abstractions (Jenner et al., 2023) have been proposed as rigorous methods to formalize and test hypotheses about how neural networks implement specific behaviors.

Causal abstraction (Geiger et al., 2023a) introduces a mathematical framework that treats both neural networks and potential explanations as causal models. An explanation is considered correct if it is a valid causal abstraction, which can be empirically tested through interchange interventions (ablations) on the neural network's activations and the explanation (Jenner et al., 2023). Various interpretability methods, such as LIME (Ribeiro et al., 2016), causal effect estimation (Feder et al., 2021), causal mediation analysis (Vig et al., 2020), iterated nullspace projection (Ravfogel et al., 2020), and circuit-based explanations are considered exceptional cases of causal abstraction (Geiger et al., 2023a). In contrast, *locally consistent abstractions* (Jenner et al., 2023) check consistency between the neural network and the explanation only one step away from the intervention node, forming a more permissive notion than causal abstraction.

Causal scrubbing (Chan et al., 2022) formalizes hypotheses as a tuple (G, I, c), where G is the model's computational graph, I is an interpretable computational graph hypothesized to explain the behavior, and c maps nodes of I to nodes of G. The core idea is to replace activations in G with other activations that should be equivalent according to the hypothesis. This is done by recursively traversing I and G, resampling important parents from the data distribution conditioned on agreeing with I, and resampling unimportant parents unconditionally. Performance is measured on the scrubbed model with resampled activations – if the hypothesis is accurate, performance should be preserved.

These methods form a hierarchy regarding strictness, with causal abstractions being the strictest, followed by locally consistent abstractions and causal scrubbing being the most permissive (Jenner et al., 2023). This hierarchy highlights trade-offs in choosing stricter or more permissive notions, affecting the ability to find acceptable explanations, generalization, and mechanistic anomaly detection. While unified by the causal framework, these methods represent different conceptual goals for what constitutes an adequate explanation of neural network behavior.

6 Current Research

This section surveys current research in mechanistic interpretability across three approaches: on architectural inductive biases for intrinsic interpretability (enhance interpretability before training) (Section 6.1), on studying phase transitions and representation emergence for developmental interpretability (interpretability during training) (Section 6.2), and on post-hoc interpretability (interpretability after training) (Section 6.3), including efforts towards uncovering general, transferable principles across models and tasks, as well as automating the discovery and interpretation of critical circuits in trained models (Section 6.4).

6.1 Intrinsic Interpretability

Intrinsic methods for mechanistic interpretability offer a promising approach to designing neural networks more amenable to reverse engineering without sacrificing performance. By encouraging *sparsity*, *modularity*, and *monosemanticity* through architectural choices and training procedures, these methods aim to make the reverse engineering process more tractable.

Intrinsic interpretability methods aim to constrain the training process to make learned programs more interpretable (Friedman et al., 2023b). This approach is closely related to neurosymbolic learning (Riegel

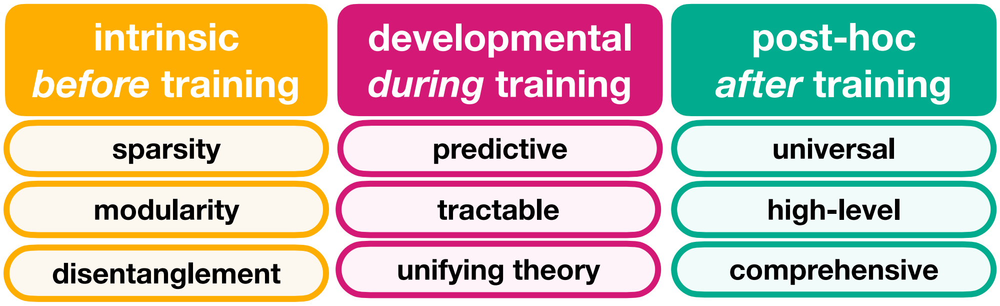


Figure 8: Key desiderata for interpretability approaches across training and analysis stages: (1) Intrinsic: Architectural biases for sparsity, modularity, and disentangled representations. (2) Developmental: Predictive capability for phase transitions, manageable number of critical transitions, and a unifying theory connecting observations to singularity geometry. (3) Post-hoc: Global, comprehensive, automated discovery of critical circuits, uncovering transferable principles across models/tasks, and extracting high-level causal mechanisms.

et al., 2020) and can involve techniques like regularization with spatial structure, akin to the organization of information in the human brain (Liu et al., 2023a;b).

Recent work has explored various architectural choices and training procedures to improve the interpretability of neural networks. Jermyn et al. (2022) and Elhage et al. (2022a) demonstrate that architectural choices can affect monosemanticity, suggesting that models could be engineered to be more monosemantic. Sharkey (2023) propose using a bilinear layer instead of a linear layer to encourage monosemanticity in language models.

Liu et al. (2023a) and Liu et al. (2023b) introduce a biologically inspired spatial regularization regime called brain-inspired modular training for forming modules in networks during training. They showcase how this can help RNNs exhibit brain-like anatomical modularity without degrading performance, in contrast to naive attempts to use sparsity to reduce the cost of having more neurons per layer (Jermyn et al., 2022; Bricken et al., 2023).

Preceding the mechanistic interpretability literature, various works have explored techniques to improve interpretability, such as sparse attention (Zhang et al., 2021), adding l_1 penalties to neuron activations (Kasioumis et al., 2021; Georgiadis, 2019), and pruning neurons (Frankle & Carbin, 2019). These techniques have been shown to encourage sparsity, modularity, and disentanglement, which are essential aspects of intrinsic interpretability.

6.2 Developmental Interpretability

Developmental interpretability focuses on learning dynamics, aiming to understand the development of internal structure in neural networks incrementally, one phase transition at a time. Singular learning theory (Watanabe, 2009; Lau et al., 2023) provides a mathematical framework for understanding the asymptotic behavior of learning algorithms in the presence of degeneracy, explaining observable effects in standard machine learning models and phenomena in deep learning, such as phase transitions.

Explaining emergence (Steinhardt, 2023; Schaeffer et al., 2023; Wei et al., 2022) and phase transitions (Simon et al., 2023) is a central theme in developmental interpretability, with phase transitions associated with mechanistic formation and changes in macroscopic behavior, such as the emergence of in-context learning (Olsson et al., 2022). The work by Hoogland et al. (2024) provides a compelling example of using the learning coefficient from singular learning theory to identify phase transitions during training that corresponded to learning bi-grams, n-grams, and induction heads in a small transformer model.

While there is currently no work directly applying developmental interpretability to explain the following phenomena, it could potentially help shed light on understanding generalization (Zhang et al., 2017), how stochastic gradient descent learns functions of increasing complexity (Nakkiran et al., 2019), and the transition from memorization to generalization (grokking) (Liu et al., 2022a; Power et al., 2022; Liu et al., 2022b; Nanda et al., 2023a; Varma et al., 2023; Thilak et al., 2022; Merrill et al., 2023; Liu et al., 2023c; Stander et al., 2023). Neural scaling laws (Caballero et al., 2022; Liu & Tegmark, 2023; Michaud et al., 2023), sometimes connected to mechanistic insights (Hernandez et al., 2022), could also potentially benefit from a developmental interpretability perspective.

In sum, developmental interpretability may serve as an evolutionary theory lens, making sense of the structures that emerge (Saphra, 2023) and offering insights into the evolution of neural network representations and their relation to learning dynamics.

6.3 Post-hoc Interpretability

In applied mechanistic interpretability, researchers explore various facets and methodologies to uncover the inner workings of AI models. Some key distinctions are drawn between *global* versus *local* interpretability and *comprehensive* versus *partial* interpretability. Global interpretability aims to uncover general patterns and behaviors of a model, providing insights that apply broadly across many instances (Doshi-Velez & Kim, 2017; Nanda, 2023e). In contrast, local interpretability explains the reasons behind a model's decisions for particular instances, offering insights into individual predictions or behaviors.

Comprehensive interpretability involves achieving a deep and exhaustive understanding of a model's behavior, providing a holistic view of its inner workings (Nanda, 2023e). In contrast, partial interpretability often applied to larger and more complex models, concentrates on interpreting specific aspects or subsets of the model's behavior, focusing on the application's most relevant or critical areas. Akin to collecting biological species, characterizing these "circuits" aims to discover general computational principles underlying modern AI systems.

This multifaceted approach collectively analyzes specific capabilities in large models while enabling a comprehensive study of learned algorithms in smaller procedural networks.

Large Models – Narrow Behavior Circuit-style mechanistic interpretability aims to explain neural networks by reverse-engineering the underlying mechanisms at the level of individual neurons or subgraphs. This approach assumes that neural vector representations encode high-level concepts and circuits defined by model weights encode meaningful algorithms (Olah et al., 2020; Cammarata et al., 2020). Studies on deep networks support these claims, identifying circuits responsible for detecting curved lines or object orientation (Cammarata et al., 2020; 2021; Voss et al., 2021).

This paradigm has been applied to language models to discover subnetworks (circuits) responsible for specific capabilities. Circuit analysis localizes and understands subgraphs within a model's computational graph responsible for specific behaviors. For large language models, this often involves narrow investigations into behaviors like multiple choice reasoning (Lieberum et al., 2023), indirect object identification (Wang et al., 2023), or computing operations (Hanna et al., 2023). Other examples include analyzing circuits for Python docstrings (Heimersheim & Jett, 2023), "an" vs "a" usage (Miller & Neo, 2023), and price tagging (Wu et al., 2023). Case studies often construct datasets using templates filled by placeholder values to enable precise control for causal interventions (Wang et al., 2023; Hanna et al., 2023; Wu et al., 2023).

Toy Models – Comprehensive Analysis Small models trained on specialized mathematical or algorithmic tasks enable more comprehensive reverse-engineering of learned algorithms (Nanda et al., 2023a; Zhong et al., 2023; Chughtai et al., 2023). Even simple arithmetic operations can involve complex strategies and multiple algorithmic solutions (Nanda et al., 2023a; Zhong et al., 2023). Characterizing these algorithms helps test hypotheses around generalizable mechanisms like variable binding (Feng & Steinhardt, 2023; Davies et al., 2023) and arithmetic reasoning (Stolfo et al., 2023). The work by Varma et al. (2023) builds on the initial grokking work and explains grokking in terms of circuit efficiency, illustrating how a comprehensive understanding of a toy model can enable interesting analyses on top of that understanding.

Towards Universality The ultimate goal is to uncover general principles that transfer across models and tasks, such as induction heads for in-context learning (Olsson et al., 2022), variable binding mechanisms

(Feng & Steinhardt, 2023; Davies et al., 2023), arithmetic reasoning (Stolfo et al., 2023; Brinkmann et al., 2024), or retrieval tasks (Variengien & Winsor, 2023). Despite promising results, debates surround the universality hypothesis – the idea that different models learn similar features and circuits when trained on similar tasks. (Chughtai et al., 2023) finds mixed evidence for universality in group composition, suggesting that while families of circuits and features can be characterized, precise circuits and development order may be arbitrary.

Towards High-level Mechanisms Causal interventions can extract a high-level understanding of computations and representations learned by large language models (Variengien & Winsor, 2023; Hendel et al., 2023; Feng & Steinhardt, 2023; Zou et al., 2023). Recent work focuses on intervening in internal representations to study high-level concepts and computations encoded. For example, Hendel et al. (2023) patched residual stream vectors to transfer task representations, while Feng & Steinhardt (2023) intervened on residual streams to argue that models generate IDs to bind entities to attributes. Representation engineering techniques (Zou et al., 2023) extract reading vectors from model activations to stimulate or inhibit specific concepts. Although these interventions don't operate via specific mechanisms, they offer a promising approach for extracting high-level causal understanding and bridging bottom-up and top-down interpretability approaches.

6.4 Automation: Scaling Post-Hoc Interpretability

As models become more complex, automating key aspects of the interpretability workflow becomes increasingly crucial. Tracing a model's computational pathways is highly labor-intensive, quickly becoming infeasible as the model size increases. Automating the discovery of relevant circuits and their functional interpretation represents a pivotal step towards scalable and comprehensive model understanding (Nainani, 2024).

Dissecting Models into Interpretable Circuits The first major automation challenge is identifying the critical computational sub-circuits or components underpinning a model's behavior for a given task. A pioneering line of work aims to achieve this via efficient **masking** or **patching** procedures. Methods like Automated Circuit Discovery (ACDC) (Conmy et al., 2023) and Attribution Patching (Syed et al., 2023; Kramár et al., 2024) iteratively knock out model activations, pinpointing components whose removal has the most significant impact on performance. This masking approach has proven scalable even to large models like Chinchilla (70B parameters) (Lieberum et al., 2023).

Other techniques take a more top-down approach. Davies et al. (2023) specify high-level causal properties (desiderata) that components solving a target subtask should satisfy and then learn binary masks to expose those component subsets. Ferrando & Voita (2024) construct Information Flow Graphs highlighting key nodes and operations by tracing attribution flows, enabling extraction of general information routing patterns across prediction domains.

Explicit architectural biases like modularity can further boost automation efficiency. Nainani (2024) find that models trained with Brain-Inspired Modular Training (BIMT) (Liu et al., 2023a) produce more readily identifiable circuits compared to standard training. Such domain-inspired inductive biases may prove increasingly vital as models grow more massive and monolithic.

Interpreting Extracted Circuits Once critical circuit components have been isolated, the key remaining step is interpreting *what* computation those components perform. Sparse autoencoders are a prominent approach for interpreting extracted circuits by decomposing neural network activations into individual component features, as discussed in Section 4.1.

A novel paradigm uses large language models themselves as an interpretive tool. Bills et al. (2023) demonstrate generating natural language descriptions of individual neuron functions by prompting language models like GPT-4 to explain sets of inputs that activate a neuron. Mousi et al. (2023) similarly employ language models to annotate unsupervised neuron clusters identified via hierarchical clustering. Foote et al. (2023) take a complementary graph-based approach in their neuron-to-graph tool: automatically extracting individual neurons' behavior patterns from training data as structured graphs amenable to visualization, programmatic comparisons, and property searches. Such representations could synergize with language model-based annotation to provide multi-faceted descriptions of neuron roles.

Other techniques map neural representations to high-level variables through gradient-based alignment. Distributed Alignment Search (DAS) (Geiger et al., 2023b) uses gradient descent to associate neuron activations with symbolic causal concepts, thereby distilling model functioning into interpretable pieces. Wu et al. (2023) scale DAS to large models like Alpaca (7B parameters) by replacing brute force steps with learned alignments.

While impressive strides have been made, robustly interpreting the largest trillion-parameter models using these techniques remains an open challenge. Another novel approach, mechanistic-interpretability-based program synthesis (Michaud et al., 2024), entirely sidesteps this complexity by auto-distilling the algorithm learned by a trained model into human-readable Python code without relying on further interpretability analyses or model architectural knowledge. As models become increasingly vast and opaque, such synergistic combinations of methods – uncovering circuits, annotating them, or altogether transcribing them into executable code – will likely prove crucial for maintaining insight and oversight.

7 Relevance

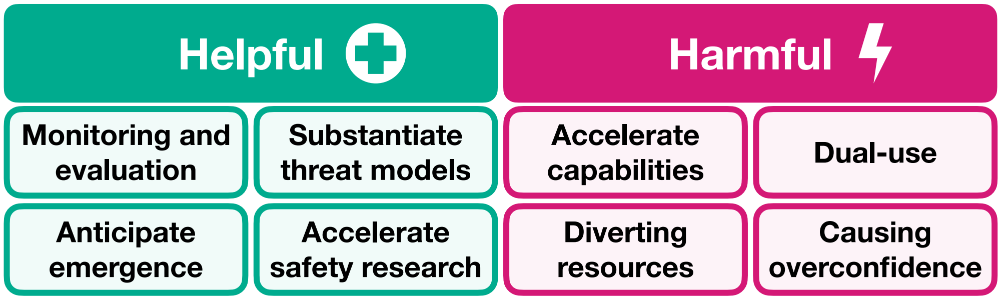


Figure 9: Potential benefits and risks of mechanistic interpretability for AI safety.

How Could Interpretability Promote AI Safety? Gaining mechanistic insights into the inner workings of AI systems seems crucial for navigating AI safety as we develop more powerful models. Interpretability tools can provide an understanding of artificial cognition, the way AI systems process information and make decisions, which offers several potential benefits:

Mechanistic interpretability could accelerate AI safety research by providing richer feedback loops and grounding for model evaluation. It may also help anticipate emergent capabilities, such as the emergence of new skills or behaviors in the model before they fully manifest. This relates to studying the incremental development of internal structures and representations as the model learns (Section 6.2). Additionally, interpretability could substantiate theoretical risk models with concrete evidence, such as demonstrating *inner misalignment* (when a model's behavior deviates from its intended goals) or *mesa-optimization* (the emergence of unintended subagents within the model). It may also trigger normative shifts within the AI community toward rigorous safety protocols by revealing potential risks or concerning behaviors.

Regarding specific AI risks, interpretability may prevent malicious misuse by locating and erasing sensitive information stored in the model. It could reduce competitive pressures by substantiating potential threats, promoting organizational safety cultures, and supporting AI alignment (ensuring AI systems pursue intended goals) through better monitoring and evaluation. Interpretability can provide safety filters for every stage of training: before training by deliberate design, during training by detecting early signs of misalignment and potentially shifting the distribution towards alignment, and after training by rigorous evaluation of artificial cognition for honesty and screening for deceptive behaviors.

Mechanistic interpretability integrates well into various AI alignment agendas, such as understanding existing models, controlling them, making AI systems solve alignment problems, and developing alignment theories. It could enhance strategies like detecting *deceptive alignment* (when a model appears aligned but is actually pursuing different goals), eliciting latent knowledge from models, and enabling better oversight. A high degree of understanding may even allow for *microscope AI* (highly interpretable AI systems) or *well-founded AI* approaches (AI systems with provable guarantees). Furthermore, comprehensive interpretability itself may be an alignment strategy if we can identify internal representations of human values and guide the model to pursue those values by retargeting an internal search process. Ultimately, understanding and control are intertwined, and better understanding can lead to improved control of AI systems.

How Could Mechanistic Insight be Harmful? Mechanistic interpretability research could accelerate AI capabilities, potentially leading to the development of powerful AI systems that are misaligned with human values, posing significant risks. While historically, interpretability research had little impact on AI capabilities, recent exceptions like discoveries about scaling laws, architectural improvements inspired by studying induction heads, and efficiency gains inspired by the logit lens technique demonstrated its potential impact. Scaling interpretability research may necessitate automation, potentially enabling rapid self-improvement of AI systems. Some researchers recommend selective publication and focusing on lower-risk areas to mitigate these risks.

Mechanistic interpretability also poses dual-use risks, where the same techniques could be used for both beneficial and harmful purposes. Fine-grained editing capabilities enabled by interpretability could be used for machine unlearning (removing private data or dangerous knowledge from models) but could be misused for censorship. Similarly, while interpretability may help improve adversarial robustness, it may also facilitate the development of stronger adversarial attacks. Knowing in advance whether interpretability research will primarily strengthen defense or offense in this domain is challenging.

Misunderstanding or overestimating the capabilities of interpretability techniques can divert resources from critical safety areas or lead to overconfidence and misplaced trust in AI systems. Robust evaluation and benchmarking (Section 9.2) are crucial to validate interpretability claims and reduce the risks of overinterpretation or misinterpretation.

8 Challenges

8.1 Research Issues

Need for Comprehensive, Multi-Pronged Approaches Current mechanistic interpretability research narrowly focuses on individual techniques rather than combining complementary approaches. Utilizing a diverse interpretability toolbox, akin to how collective innovations like batch normalization, residual connections, and improved optimizers drove advances in computer vision, could provide holistic understanding, e.g., coordinated methods reverse-engineering trojaned behaviors (Casper et al., 2023b).

There is an overemphasis on post-hoc techniques, with fewer efforts on intrinsic approaches enhancing interpretability through architectural inductive biases or training (Sharkey, 2023; Elhage et al., 2022a; Wong et al., 2023; Hubinger, 2019c). Suggesting intrinsic interpretability complements post-hoc analysis for robust understanding. More work predicting and shaping capabilities in advance, rather than merely explaining afterward, could benefit the field.

Cherry-Picking and Streetlight Interpretability. Another concerning pattern is the tendency to cherry-pick results, relying on a small number of convincing examples or visualizations as the basis for an argument without comprehensive evaluation (Räuker et al., 2023). This amounts to publication bias, showcasing an unrealistic highlight reel of best-case performance. Relatedly, many interpretability techniques are primarily evaluated on small toy models and tasks (Chughtai et al., 2023; Elhage et al., 2022b; Jermyn et al., 2022; Chen et al., 2023b), risking missing critical phenomena that only emerge in more realistic and diverse contexts. This focus on cherry-picked results from toy models is a form of "streetlight interpretability" (Casper, 2023), examining AI systems under only ideal conditions of maximal interpretability.

8.2 Technical Limitations

Scalability Challenges and Risks of Human Reliance. A critical hurdle is demonstrating the scalability of mechanistic interpretability to real-world AI systems across model size, task complexity, behavioral coverage, and analysis efficiency (Elhage et al., 2022b; Scherlis et al., 2023). Achieving a truly comprehensive understanding of a model's capabilities in all contexts is daunting, and the time and compute required must scale tractably. Automating interpretability techniques is crucial, as manual analysis quickly becomes infeasible for large models. The high human involvement in current interpretability research raises concerns about the scalability and validity of human-generated model interpretations. Subjective, inconsistent human evaluations and lack of ground-truth benchmarks are known issues (Räuker et al., 2023). As models scale, it will become increasingly untenable to rely on humans to hypothesize about model mechanisms manually. More work is needed on automating the discovery of mechanistic explanations and translating model weights into human-readable computational graphs (Elhage et al., 2022b).

Obstacles to Bottom-Up Interpretability. There are fundamental questions about the tractability of fully reverse-engineering neural networks from the bottom up, especially as models become more complex (Hendrycks, 2023). Models may learn internal representations and algorithms that do not cleanly map to human-understandable concepts, making them difficult to interpret even with complete transparency (McGrath et al., 2022). This gap between human and model ontologies may widen as architectures evolve, increasing opaqueness (Hendrycks et al., 2022). Conversely, model representations might naturally converge to more human-interpretable forms as capability increases (Hubinger, 2019a; Feng & Steinhardt, 2023).

Analyzing Models Embedded in Environments Real-world AI systems embedded in rich, interactive environments exhibit two forms of in-context behavior that pose significant interpretability challenges beyond understanding models in isolation. Externally, models may dynamically adapt to and reshape their environments through in-context learning from the interactions and feedback loops with their external environment (Leahy, 2023). Internally, the Hydra effect demonstrates in-context reorganization, where models flexibly reorganize their internal representations in a context-dependent manner to maintain capabilities even after ablating key components (McGrath et al., 2023). These two instances of in-context behavior - external adaptation to the environment and internal self-reorganization - undermine interpretability approaches that assume fixed circuits. For models deeply embedded in rich real-world settings, their dynamic coupling with the external world via in-context environmental learning and their internal in-context representational reorganization make strong interpretability guarantees difficult to attain through analysis of the initial model alone.

Adversarial Pressure Against Interpretability As models become more capable through increased training and optimization, there is a risk they may learn deceptive behaviors that actively obscure or mislead the interpretability techniques meant to understand them. Models could develop adversarial "mind-reader" components that predict and counteract the specific analysis methods used to interpret their inner workings (Sharkey, 2022; Hubinger, 2022). Optimizing models through techniques like gradient descent could inadvertently make their internal representations less interpretable to external observers (Hubinger, 2019b; Fu et al., 2023; von Oswald et al., 2023). In extreme cases, a highly advanced AI system singularly focused on preserving its core objectives may directly undermine the fundamental assumptions that enable interpretability methods in the first place.

These adversarial dynamics, where the capabilities of the AI model are pitted against efforts to interpret it, underscore the need for interpretability research to prioritize worst-case robustness rather than just average-case scenarios. Current techniques often fail even when models are not adversarially optimized. Achieving high confidence in fully understanding extremely capable AI models may require fundamental advances to make interpretability frameworks resilient against an intelligent system's active deceptive efforts.

9 Future Directions

Given the current limitations and challenges, several promising directions can be pursued to advance mechanistic interpretability, emphasizing conceptual clarity, establishing rigorous standards, improving the scalability of interpretability techniques, and expanding the research scope.

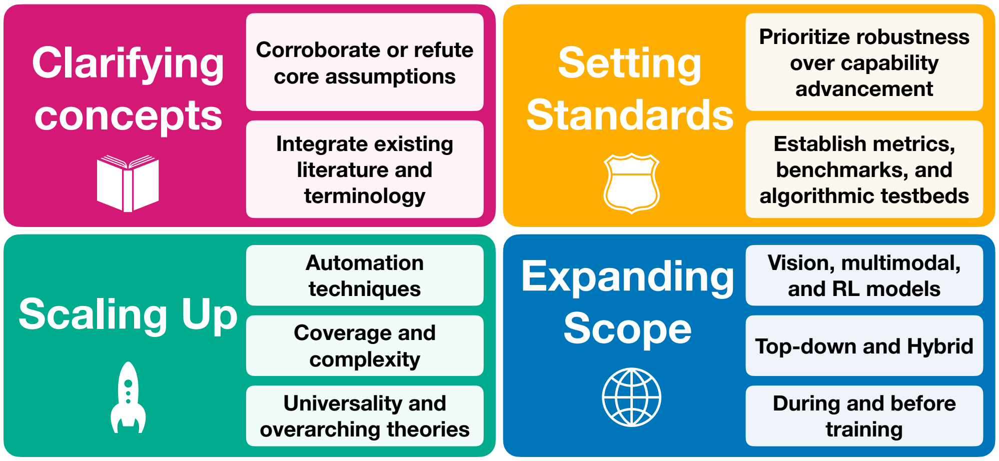

Clarifying concepts
Corroborate or refute core assumptions
Integrate existing literature and terminology
Setting Standards
Prioritize robustness over capability advancement
Establish metrics, benchmarks, and algorithmic testbeds
Scaling Up
Automation techniques
Coverage and complexity
Universality and overarching theories
Expanding Scope
Vision, multimodal, and RL models
Top-down and Hybrid
During and before training

Figure 10: Roadmap for advancing mechanistic interpretability research, highlighting key strategic directions.

9.1 Clarifying Concepts

Integrating with Existing Literature. To mature, mechanistic interpretability should embrace existing work, using established terminology rather than reinventing the wheel. Diverging terminology inhibits collaboration across disciplines. Presently, the terminology used for mechanistic interpretability partially diverges from mainstream AI research (Casper, 2023). For example, while the mainstream speaks of *distributed representations* (Hinton, 1984; Olah, 2023) and the goal of *disentangling representations* (Higgins et al., 2018; Locatello et al., 2019), the mechanistic interpretability literature refers to the same phenomenon as *polysemanticity* (Scherlis et al., 2023; Lecomte et al., 2023; Marshall & Kirchner, 2024) and *superposition* (Elhage et al., 2022b; Henighan et al., 2023). Using common language invites "accidental" contributions and prevents isolating mechanistic interpretability from broader AI research.

Mechanistic interpretability relates to many other fields in AI research, including compressed sensing (Elhage et al., 2022b), modularity, adversarial robustness, continual learning, network compression (Räuker et al., 2023), neurosymbolic reasoning, trojan detection, and program synthesis (Casper, 2023; Michaud et al., 2024). These relationships can help develop new methods, metrics, benchmarks, and theoretical frameworks. For instance:

- **Neurosymbolic Reasoning and Program Synthesis**: Mechanistic interpretability aims to *reverse engineer* neural networks by converting their weights into human-readable algorithms. This endeavor can draw inspiration from neurosymbolic reasoning (Riegel et al., 2020) and program synthesis. Techniques like creating programs in domain-specific languages (Verma et al., 2019b;a; Trivedi et al., 2021), extracting decision trees (Zhang et al., 2019) or symbolic causal graphs (Ren et al., 2023) from neural networks align well with the goals of mechanistic interpretability. Adopting these approaches can extend the toolkit for reverse engineering AI systems.
- **Trojan Detection**: Detecting deceptive models is a key motivation for inspecting model internals, as deception is not salient from observing behavior alone by definition (Casper et al., 2024). However, quantifying progress is challenging due to the lack of evidence for deception as an emergent capability in current models (Steinhardt, 2023), apart from sycophancy (Sharma et al., 2023) and theoretical evidence for deceptive inflation behavior (Lang et al., 2024). Detecting trojans or

backdoors (Hubinger et al., 2024) implanted via data poisoning could serve as a helpful proxy goal and proof-of-concept. While these trojans simulate outer alignment failure (misalignment between the model's behavior and its specified objective) rather than inner alignment failure like deceptive alignment (where an emergent sub-component optimizer within the model is misaligned with the original training objective), trojan detection still provides a practical testbed for benchmarking interpretability methods and evaluating their effectiveness quantitatively.

- **Adversarial Robustness**: There is a duality between interpretability and adversarial robustness (Elhage et al., 2022b; Räuker et al., 2023). More interpretable models tend to be more robust against adversarial attacks. Interpretability tools can help create more sophisticated adversaries, improving our understanding of model internals. Viewing adversarial examples as inherent neural network *features* (Ilyas et al., 2019) rather than bugs also hints at alien features beyond human perception. Connecting mechanistic interpretability to adversarial robustness thus offers paths to gain theoretical insights and measure progress, for instance, by evaluating how well interpretability enables crafting strong adversarial examples.

More details on the interplay between interpretability, robustness, modularity, continual learning, network compression, and the human visual system can be found in the review by Räuker et al. (2023).

Corroborate or Refute Core Assumptions. Features are the fundamental units defining neural representations and enabling mechanistic interpretability's bottom-up approach (Chan, 2023), but defining them involves assumptions requiring scrutiny, as they shape interpretations and research directions. Questioning hypotheses by seeking additional evidence or counter-examples is crucial.

The **linear representation hypothesis** treats activation directions as features (Park et al., 2023; Nanda et al., 2023b; Elhage et al., 2022b), but the emergence and necessity of linearity is unclear - is it architectural bias or inherent? Stronger theory justifying linearity's necessity or counter-examples like autoencoders on uncorrelated data without intermediate linear layers (Elhage et al., 2022b) are needed. An alternative lens views features as polytopes from piecewise linear activations (Black et al., 2022), questioning if direction simplification suffices or added polytope complexity aids interpretability.

Polysemantic neurons are attributed to **superposition** compressing many features into limited neurons (Elhage et al., 2022b), but incidental redundancy without compression also causes polysemanticity (Lecomte et al., 2023; Marshall & Kirchner, 2024; McGrath et al., 2023). Understanding superposition's role could inform mitigating polysemanticity via regularization (Lecomte et al., 2023). Superposition also raises open questions like operationalizing computation in superposition (Vaintrob et al., 2024), attention head superposition (Elhage et al., 2022b; Jermyn et al., 2023; Lieberum et al., 2023; Gould et al., 2023), representing feature clusters (Elhage et al., 2022b), connections to adversarial robustness (Elhage et al., 2022b), anti-correlated feature organization (Elhage et al., 2022b), and architectural effects (Nanda, 2023a).

9.2 Setting Standards

Prioritizing Robustness over Capability Advancement. As the interpretability community expands, it is essential to maintain the norm of not advancing AI capabilities while simultaneously establishing metrics necessary for the field's progress (Räuker et al., 2023). Researchers should prioritize developing comprehensive tools for analyzing the worst-case performance of AI systems, ensuring robustness and reliability in critical applications. This includes focusing on adversarial tasks, such as backdoor detection and removal (Lamparth & Reuel, 2023; Hubinger et al., 2024; Wu et al., 2022), and evaluating the accuracy of explanations in producing adversarial examples (Goldowsky-Dill et al., 2023).

Establishing Metrics, Benchmarks, and Algorithmic Testbeds. To objectively evaluate interpretability methods, developing well-defined metrics and standardized benchmarks assessing aspects like feature attribution accuracy, circuit explanation comprehensiveness, and practical utility across diverse models/tasks is crucial (Räuker et al., 2023). Algorithmic testbeds are also essential for evaluating faithfulness (Jacovi & Goldberg, 2020) and falsifiability (Leavitt & Morcos, 2020) of techniques. Tools like Tracr (Lindner et al., 2023) can provide ground truth labels for benchmarking search methods (Goldowsky-Dill et al., 2023). Toy models studying **superposition in computation** (Vaintrob et al., 2024), transformers on algorithmic tasks can quantify sparsity and test intrinsic methods. Replacing components with hypothesized circuits (Quirke et al., 2024) should be the goal for comprehensive evaluation.

9.3 Scaling Up

Broader and Deeper Coverage of Complex Models and Behaviors. A primary goal in scaling mechanistic interpretability is pushing the Pareto frontier between model and task complexity and the coverage of interpretability techniques (Chan, 2023). While efforts have focused on larger models, it is equally crucial to scale to more complex tasks and provide comprehensive explanations essential for provable safety (Tegmark & Omohundro, 2023) and enumerative safety (Cunningham et al., 2024; Elhage et al., 2022b) by ensuring models won't engage in dangerous behaviors like deception. Future work should aim for thorough reverse-engineering (Quirke & Barez, 2023), integrating proven modules into larger networks (Nanda et al., 2023a), and capturing sequences encoded in hidden states beyond immediate predictions (Pal et al., 2023). Deepening analysis complexity is also key, validating the realism of toy models (Elhage et al., 2022b) and extending techniques like path patching (Goldowsky-Dill et al., 2023; Liu et al., 2023a) to larger language models. To tackle more complex, realistic cases, the field must move beyond small transformers on algorithmic tasks (Nanda et al., 2023a) and limited scenarios (Friedman et al., 2023a).

Towards Universality. As interpretability matures, the field must transition from isolated empirical findings to developing overarching theories and universal reasoning primitives beyond specific circuits, aiming for a comprehensive understanding of AI capabilities. While collecting empirical data remains valuable (Nanda, 2023f), establishing motifs, empirical laws, and theories capturing universal model behavior aspects is crucial. This may involve finding more circuits/features (Nanda, 2022a;c), exploring circuits as a lens for memorization/generalization (Hanna et al., 2023), identifying primitive general reasoning skills (Feng & Steinhardt, 2023), generalizing specific findings to model-agnostic phenomena (Merullo et al., 2023), and investigating emergent model generality across neural network classes (Ivanitskiy et al., 2023). Identifying universal reasoning patterns and unifying theories is key to advancing interpretability.

Automation. Implementing automated methods is crucial for scaling interpretability of real-world state-of-the-art models across size, task complexity, behavior coverage, and analysis time (Hobbhahn, 2022). Manual circuit identification is labor-intensive (Lieberum et al., 2023), so automated techniques like circuit discovery and sparse autoencoders can enhance the process (Foote et al., 2023; Nanda, 2023b). Future work should automatically create varying datasets for understanding circuit functionality (Conmy et al., 2023), develop automated hypothesis search (Goldowsky-Dill et al., 2023), and investigate attention head/MLP interplay (Monea et al., 2023). Scaling sparse autoencoders to extract high-quality features automatically for frontier models is critical (Bricken et al., 2023). Still, it requires caution regarding potential downsides like AI iteration outpacing training (___RicG___, 2023) and loss of human interpretability from tool complexity (Doshi-Velez & Kim, 2017).

9.4 Expanding the Scope

Interpretability Across Training While mechanistic interpretability of final trained models is a prerequisite, the field should also advance interpretability before and during training by studying learning dynamics (Nanda, 2022b; Elhage et al., 2022b; Hubinger, 2022). This includes tracking neuron development (Liu et al., 2021), analyzing neuron set changes with scale (Michaud et al., 2023), and investigating emergent computations (Quirke & Barez, 2023). Studying phase transitions could yield safety insights like reward hacking risks (Olsson et al., 2022).

Multi-Level Analysis Complementing the predominant bottom-up methods (Hanna et al., 2023), mechanistic interpretability should explore top-down and hybrid approaches, a promising yet neglected avenue. The top-down analysis offers a tractable way to study large models and guide microscopic research with macroscopic observations (Variengien & Winsor, 2023). Its computational efficiency could enable extensive "comparative anatomy" of diverse models, revealing high-level motifs underlying abilities. These motifs could serve as analysis units for understanding internal modifications from techniques like instruction fine-tuning (Ouyang et al., 2022) and reinforcement learning from human feedback (Christiano et al., 2017; Bai et al., 2022).

New Frontiers: Vision, Multimodal, and Reinforcement Learning Models While some mechanistic interpretability has explored convolutional neural networks for vision (Cammarata et al., 2021; 2020), vision-language models (Palit et al., 2023; Salin et al., 2022; Hilton et al., 2020), and multimodal neurons (Goh et al., 2021), little work has focused on vision transformers (Palit et al., 2023; Aflalo et al., 2022; Vilas et al., 2023). Future efforts could identify mechanisms within vision-language models, mirroring progress in unimodal language models (Nanda et al., 2023a; Wang et al., 2023).

Reinforcement learning (RL) is also a crucial frontier given its role in advanced AI training via techniques like reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Bai et al., 2022), despite potentially posing significant safety risks (Bereska & Gavves, 2023; Casper et al., 2023a). Interpretability of RL should investigate reward/goal representations (TurnTrout et al., 2023; Colognese & Jozdien, 2023; Colognese, 2023; Bloom & Colognese, 2023), study circuitry changes from alignment algorithms (Jain et al., 2023; Lee et al., 2024), and explore emergent subgoals or proxies (Hubinger et al., 2019; Ivanitskiy et al., 2023).

References

Estelle Aflalo, Meng Du, Shao-Yen Tseng, Yongfei Liu, Chenfei Wu, Nan Duan, and Vasudev Lal. VL-InterpreT: An Interactive Visualization Tool for Interpreting Vision-Language Transformers. *CVPR*, June 2022. 27

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *ICLR*, 2016. 7

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear Algebraic Structure of Word Senses, with Applications to Polysemy. *TACL*, December 2018. 4

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, July 2015. 2

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *CoRR*, April 2022. 27

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. GAN Dissection: Visualizing and Understanding Generative Adversarial Networks. *ICLR*, December 2018. 16

Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *CoRR*, September 2021. 2, 7, 11

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting Latent Predictions from Transformers with the Tuned Lens. *CoRR*, August 2023. 12

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *ACM FAccT*, March 2021. 9

Leonard Bereska and Efstratios Gavves. Taming Simulators: Challenges, Pathways and Vision for the Alignment of Large Language Models. *AAAI Symposium Series*, October 2023. 10, 27

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *OpenAI Blog*, 2023. 3, 20

Christopher M. Bishop. *Pattern recognition and machine learning*. Springer-Verlag New York Inc., 2006. 3

Sid Black, Lee Sharkey, Leo Grinsztajn, Eric Winsor, Dan Braun, Jacob Merizian, Kip Parker, Carlos Ramón Guevara, Beren Millidge, Gabriel Alfour, and Connor Leahy. Interpreting Neural Networks through the Polytope Lens. *CoRR*, November 2022. 7, 25

Joseph Bloom and Paul Colognese. Decision Transformer Interpretability. *AI Alignment Forum*, 2023. 27

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Vi'egas, and M. Wattenberg. An Interpretability Illusion for BERT. *CoRR*, April 2021. 16

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*, October 2023. 4, 6, 7, 8, 13, 18, 26

Jannik Brinkmann, Abhay Sheshadri, Victor Levoso, Paul Swoboda, and Christian Bartelt. A Mechanistic Analysis of a Transformer Trained on a Symbolic Multi-Step Reasoning Task. *CoRR*, February 2024. 20

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *CoRR*, April 2023. 1

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering Latent Knowledge in Language Models Without Supervision. *ICLR*, 2023. 2, 11, 16

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken Neural Scaling Laws. *ICLR*, October 2022. 19

Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve Detectors. *Distill*, June 2020. 8, 19, 27

Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah. Curve Circuits. *Distill*, 2021. 8, 19, 27

Steven Cao, Victor Sanh, and Alexander M. Rush. Low-Complexity Probing via Finding Subnetworks. *NAACL-HLT*, April 2021. 11

Giuseppe Casalicchio, Christoph Molnar, and Bernd Bischl. Visualizing the Feature Importance for Black Box Models. *ECML PKDD*, 2018. 2

Stephen Casper. The Engineer's Interpretability Sequence. *AI Alignment Forum*, February 2023. 16, 22, 24

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Bıyık, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback. *CoRR*, 2023a. 10, 27

Stephen Casper, Yuxiao Li, Jiawei Li, Tong Bu, Kevin Zhang, Kaivalya Hariharan, and Dylan Hadfield-Menell. Red Teaming Deep Neural Networks with Feature Synthesis Tools. *NeurIPS*, 2023b. 16, 22

Stephen Casper, Carson Ezell, Charlotte Siegmann, Noam Kolt, Taylor Lynn Curtis, Benjamin Bucknall, Andreas Haupt, Kevin Wei, Jérémy Scheurer, Marius Hobbhahn, Lee Sharkey, Satyapriya Krishna, Marvin Von Hagen, Silas Alberti, Alan Chan, Qinyi Sun, Michael Gerovitch, David Bau, Max Tegmark, David Krueger, and Dylan Hadfield-Menell. Black-Box Access is Insufficient for Rigorous AI Audits. *CoRR*, January 2024. 24

Lawrence Chan. What I would do if I wasn't at ARC Evals. *AI Alignment Forum*, May 2023. 25, 26

Lawrence Chan, Adrià Garriga-alonso, Nicholas Goldowsky-Dill, ryan_greenblatt, jenny, Ansh Radhakrishnan, Buck, and Nate Thomas. Causal Scrubbing: a method for rigorously testing interpretability hypotheses [Redwood Research]. *AI Alignment Forum*, December 2022. 17

Lawrence Chan, Leon Lang, and Erik Jenner. Natural Abstractions: Key claims, Theorems, and Critiques. *AI Alignment Forum*, March 2023. 8

David Chanin, Anthony Hunter, and Oana-Maria Camburu. Identifying Linear Relational Concepts in Large Language Models. *CoRR*, 2023. 7

Yiting Chen, Zhanpeng Zhou, and Junchi Yan. Going Beyond Neural Network Feature Similarity: The Network Feature Complexity and Its Interpretation Using Category Theory. *CoRR*, November 2023a. 8

Zhongtian Chen, Edmund Lau, Jake Mendel, Susan Wei, and Daniel Murfet. Dynamical versus Bayesian Phase Transitions in a Toy Model of Superposition. *CoRR*, October 2023b. 22

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *NeurIPS*, December 2017. 27

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A Toy Model of Universality: Reverse Engineering How Networks Learn Group Operations. *ICML*, 2023. 9, 19, 20, 22

Paul Colognese. Internal Target Information for AI Oversight. *LessWrong*, 2023. 27

Paul Colognese and Jozdien. High-level interpretability: detecting an AI's objectives. *AI Alignment Forum*, 2023. 27

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards Automated Circuit Discovery for Mechanistic Interpretability. *NeurIPS*, 2023. 15, 20, 26

Ian C. Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: a unified framework for model explanation. *JMLR*, January 2021. 2

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models. *ICLR*, January 2024. 6, 7, 12, 13, 26

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge Neurons in Pretrained Transformers. *ACL*, 2022. 3, 16

James Dao, Yeu-Tong Lau, Can Rager, and Jett Janiak. An Adversarial Example for Direct Logit Attribution: Memory Management in gelu-4l. *CoRR*, 2023. 15

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing Transformers in Embedding Space. *ACL*, December 2022. 12

Xander Davies, Max Nadeau, Nikhil Prakash, Tamar Rott Shaham, and David Bau. Discovering Variable Binding Circuitry with Desiderata. *CoRR*, July 2023. 8, 19, 20

Mingyang Deng, Lucas Tao, and Joe Benton. Measuring Feature Sparsity in Language Models. *CoRR*, 2023. 7, 13

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. Jump to Conclusions: Short-Cutting Transformers With Linear Transformations. *CoRR*, March 2023. 12

Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. *CoRR*, March 2017. 16, 19, 26

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. Analyzing Individual Neurons in Pretrained Language Models. *EMNLP*, October 2020. 3

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *TACL*, February 2021. 11

Nelson Elhage, Tristan Hume, Olsson Catherine, Nanda Neel, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. Softmax Linear Units. *Transformer Circuits Thread*, 2022a. 4, 6, 18, 22

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and others. Toy Models of Superposition. *Transformer Circuits Thread*, 2022b. 3, 4, 5, 6, 12, 22, 23, 24, 25, 26

Sebastian Farquhar, Vikrant Varma, Zachary Kenton, Johannes Gasteiger, Vladimir Mikulik, and Rohin Shah. Challenges with unsupervised LLM knowledge discovery. *CoRR*, 2023. 12

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. CausaLM: Causal Model Explanation Through Counterfactual Language Models. *Computational Linguistics*, May 2021. 17

Jiahai Feng and Jacob Steinhardt. How do Language Models Bind Entities in Context? *CoRR*, October 2023. 8, 19, 20, 23, 26

Javier Ferrando and Elena Voita. Information Flow Routes: Automatically Interpreting Language Models at Scale. *CoRR*, February 2024. 20

Alex Foote, Neel Nanda, Esben Kran, Ioannis Konstas, Shay Cohen, and Fazl Barez. Neuron to Graph: Interpreting Language Model Neurons at Scale. *CoRR*, May 2023. 20, 26

Jonathan Frankle and Michael Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. *ICLR*, March 2019. 18

Dan Friedman, Andrew Lampinen, Lucas Dixon, Danqi Chen, and Asma Ghandeharioun. Interpretability illusions in the generalization of simplified models. *CoRR*, 2023a. 26

Dan Friedman, Alexander Wettig, and Danqi Chen. Learning Transformer Programs. *NeurIPS*, June 2023b. 17

Deqing Fu, Tian-Qi Chen, Robin Jia, and Vatsal Sharan. Transformers Learn Higher-Order Optimization Methods for In-Context Learning: A Study with Linear Models. *CoRR*, October 2023. 23

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal Abstractions of Neural Networks. *NeurIPS*, 2021a. 17

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah D. Goodman, and Christopher Potts. Inducing Causal Structure for Interpretable Neural Networks. *ICML*, January 2021b. 13, 17

Atticus Geiger, Chris Potts, and Thomas Icard. Causal Abstraction for Faithful Model Interpretation. *CoRR*, January 2023a. 16, 17

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. Finding Alignments Between Interpretable Causal Variables and Distributed Neural Representations. *CoRR*, 2023b. 17, 21

Georgios Georgiadis. Accelerating Convolutional Neural Networks via Activation Map Compression. *CoRR*, March 2019. 18

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting Recall of Factual Associations in Auto-Regressive Language Models. *EMNLP*, October 2023. 14

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A Unifying Framework for Inspecting Hidden Representations of Language Models. *CoRR*, January 2024. 16

Amirata Ghorbani and James Zou. Neuron Shapley: Discovering the Responsible Neurons. *NeurIPS*, November 2020. 3, 16

Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal Neurons in Artificial Neural Networks. *Distill*, March 2021. 3, 27

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing Model Behavior with Path Patching. *CoRR*, 2023. 14, 16, 25, 26

Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. Successor Heads: Recurring, Interpretable Attention Heads In The Wild. *CoRR*, 2023. 25

Wes Gurnee and Max Tegmark. Language Models Represent Space and Time. *CoRR*, 2023. 9

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding Neurons in a Haystack: Case Studies with Sparse Probing. *TMLR*, 2023. 6, 11

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal Neurons in GPT2 Language Models. *CoRR*, January 2024. 9

David R. Ha and J. Schmidhuber. Recurrent World Models Facilitate Policy Evolution. *NeurIPS*, September 2018. 9

Guy Hacohen, Leshem Choshen, and Daphna Weinshall. Let's Agree to Agree: Neural Networks Share Classification Order on Real Datasets. *ICML*, 2020. 8

Michael Hanna, Ollie Liu, and Alexandre Variengien. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *NeurIPS*, 2023. 8, 14, 19, 26

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does Localization Inform Editing? Surprising Differences in Causality-Based Localization vs. Knowledge Editing in Language Models. *NeurIPS Spotlight*, January 2023. 15, 16

Stefan Heimersheim and Jett. A circuit for Python docstrings in a 4-layer attention-only transformer. *AI Alignment Forum*, February 2023. 8, 19

Roee Hendel, Mor Geva, and Amir Globerson. In-Context Learning Creates Task Vectors. *EMNLP*, October 2023. 7, 14, 20

Dan Hendrycks. *Introduction to AI Safety, Ethics, and Society*. Self-published, 2023. 23

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved Problems in ML Safety. *CoRR*, June 2022. 23

Tom Henighan, Shan Carter, Tristan Hume, Nelson Elhage, Robert Lasenby, Stanislav Fort, Nicholas Schiefer, and Christopher Olah. Superposition, Memorization, and Double Descent. *Transformer Circuits Thread*, 2023. 6, 24

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. Scaling Laws and Interpretability of Learning from Repeated Data. *CoRR*, 2022. 19

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of Relation Decoding in Transformer Language Models. *CoRR*, August 2023. 7

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a Definition of Disentangled Representations. *CoRR*, December 2018. 24

Jacob Hilton, Nick Cammarata, Shan Carter, Gabriel Goh, and Chris Olah. Understanding RL Vision. *Distill*, 2020. 27

Geoffrey E Hinton. Distributed representations. *Carnegie Mellon University*, 1984. 24

Marius Hobbhahn. Marius' alignment agenda, 2022. 26

Jesse Hoogland, George Wang, Matthew Farrugia-Roberts, Liam Carroll, Susan Wei, and Daniel Murfet. The Developmental Landscape of In-Context Learning. *CoRR*, February 2024. 18

Jing Huang, Atticus Geiger, Karel D'Oosterlinck, Zhengxuan Wu, and Christopher Potts. Rigorously Assessing Natural Language Explanations of Neurons. *CoRR*, September 2023. 3

Evan Hubinger. Chris Olah's views on AGI safety. *AI Alignment Forum*, November 2019a. 3, 23

Evan Hubinger. Gradient hacking. *AI Alignment Forum*, October 2019b. 23

Evan Hubinger. Relaxed adversarial training for inner alignment. *AI Alignment Forum*, September 2019c. 22

Evan Hubinger. A transparency and interpretability tech tree. *AI Alignment Forum*, June 2022. 7, 23, 26

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from Learned Optimization in Advanced Machine Learning Systems. *CoRR*, May 2019. 27

Evan Hubinger, Adam Jermyn, Johannes Treutlein, Rubi Hudson, and Kate Woolverton. Conditioning Predictive Models: Risks and Strategies. *CoRR*, February 2023. 9, 10

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. *CoRR*, 2024. 25

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial Examples Are Not Bugs, They Are Features. *NeurIPS*, August 2019. 3, 16, 25

M. Ivanitskiy, Alexander F. Spies, Tilman Rauker, Guillaume Corlouer, Chris Mathwin, Lucia Quirke, Can Rager, Rusheb Shah, Dan Valentine, Cecilia Diniz Behn, Katsumi Inoue, and Samy Wu Fung. Structured World Representations in Maze-Solving Transformers. *CoRR*, December 2023. 9, 26, 27

Alon Jacovi and Yoav Goldberg. Towards Faithfully Interpretable NLP Systems: How should we define and evaluate faithfulness? *CoRR*, April 2020. 25

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. *CoRR*, November 2023. 27

janus. Simulators. *LessWrong*, September 2022. 9, 10

Erik Jenner, Adrià Garriga-alonso, and Egor Zverev. A comparison of causal scrubbing, causal abstractions, and related methods. *AI Alignment Forum*, June 2023. 17

Adam Jermyn, Chris Olah, and T Henighan. Circuits updates - May 2023: Attention Head Superposition. *Transformer Circuits Thread*, 2023. 25

Adam S. Jermyn, Nicholas Schiefer, and Evan Hubinger. Engineering Monosemanticity in Toy Models. *CoRR*, November 2022. 6, 18, 22

Jozdien. Conditioning Generative Models for Alignment. *AI Alignment Forum*, July 2022. 9

Jaap Jumelet. Evaluating and Interpreting Language Models. *NLP Lecture*, November 2023. 2

Theodoros Kasioumis, Joe Townsend, and Hiroya Inakoshi. Elite BackProp: Training Sparse Interpretable Neurons. *International Workshop on Neuro-Symbolic Learning and Reasoning*, 2021. 18

Jan Kirchner. Neuroscience and Natural Abstractions. *LessWrong*, March 2023. 8

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. *ICML*, July 2019. 8

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. AtP*: An efficient and scalable method for localizing LLM behaviour to components. *CoRR*, March 2024. 16, 20

Jan Kulveit, Clem von Stengel, and Roman Leventov. Predictive Minds: LLMs As Atypical Active Inference Agents. *CoRR*, November 2023. 9

Max Lamparth and Anka Reuel. Analyzing And Editing Inner Mechanisms Of Backdoored Language Models. *CoRR*, 2023. 25

Michael Lan and Fazl Barez. Locating Cross-Task Sequence Continuation Circuits in Transformers. *CoRR*, November 2023. 8

Leon Lang, Davis Foote, Stuart Russell, Anca Dragan, Erik Jenner, and Scott Emmons. When Your AIs Deceive You: Challenges with Partial Observability of Human Evaluators in Reward Learning. *CoRR*, March 2024. 24

Georg Lange, Alex Makelov, and Neel Nanda. An Interpretability Illusion for Activation Patching of Arbitrary Subspaces. *AI Alignment Forum*, August 2023. 15

Anna Langedijk, Hosein Mohebbi, Gabriele Sarti, Willem Zuidema, and Jaap Jumelet. DecoderLens: Layerwise Interpretation of Encoder-Decoder Transformers. *CoRR*, 2023. 12

Edmund Lau, Daniel Murfet, and Susan Wei. Quantifying degeneracy in singular models via the learning coefficient. *CoRR*, August 2023. 18

Connor Leahy. Barriers to Mechanistic Interpretability for AGI Safety. *AI Alignment Forum*, 2023. 23

Matthew L. Leavitt and Ari Morcos. Towards falsifiable interpretability research. *CoRR*, October 2020. 17, 25

Victor Lecomte, Kushal Thaman, Trevor Chow, Rylan Schaeffer, and Sanmi Koyejo. Incidental Polysemanticity. *CoRR*, 2023. 24, 25

Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K. Kummerfeld, and Rada Mihalcea. A Mechanistic Understanding of Alignment Algorithms: A Case Study on DPO and Toxicity. *CoRR*, 2024. 27

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. *NeurIPS*, 2006. 12

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. *ICLR*, 2023a. 7, 9, 11

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. *NeurIPS Spotlight*, July 2023b. 7

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent Learning: Do different neural networks learn the same representations? *NIPS Workshop on Feature Extraction*, December 2015. 8

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does Circuit Analysis Interpretability Scale? Evidence from Multiple Choice Capabilities in Chinchilla. *CoRR*, July 2023. 8, 14, 19, 20, 25, 26

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Thomas McGrath, and Vladimir Mikulik. Tracr: Compiled Transformers as a Laboratory for Interpretability. *CoRR*, 2023. 16, 25

Leo Z. Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. Probing Across Time: What Does RoBERTa Know and When? *EMNLP*, September 2021. 26

Ziming Liu and Max Tegmark. A Neural Scaling Law from Lottery Ticket Ensembling. *CoRR*, 2023. 19

Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J. Michaud, Max Tegmark, and Mike Williams. Towards Understanding Grokking: An Effective Theory of Representation Learning. *NeurIPS*, 2022a. 19

Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking Beyond Algorithmic Data. *ICML*, 2022b. 19

Ziming Liu, Eric Gan, and Max Tegmark. Seeing is Believing: Brain-Inspired Modular Training for Mechanistic Interpretability. *Entropy*, June 2023a. 18, 20, 26

Ziming Liu, Mikail Khona, Ila R. Fiete, and Max Tegmark. Growing Brains: Co-emergence of Anatomical and Functional Modularity in Recurrent Neural Networks. *CoRR*, 2023b. 18

Ziming Liu, Ziqian Zhong, and Max Tegmark. Grokking as compression: A nonlinear complexity perspective. *CoRR*, 2023c. 19

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. *CoRR*, June 2019. 24

Luke Marks, Amir Abdullah, Luna Mendez, Rauno Arike, Philip Torr, and Fazl Barez. Interpreting Reward Models in RLHF-Tuned Language Models Using Sparse Autoencoders. *CoRR*, October 2023. 13

Simon C. Marshall and Jan H. Kirchner. Understanding polysemanticity in neural networks through coding theory. *CoRR*, January 2024. 4, 24, 25

Mantas Mazeika, Andy Zou, Akul Arora, Pavel Pleskov, Dawn Song, Dan Hendrycks, Bo Li, and David Forsyth. How Hard is Trojan Detection in DNNs? Fooling Detectors With Evasive Trojans. *CoRR*, September 2022. 16

Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. Copy Suppression: Comprehensively Understanding an Attention Head. *CoRR*, October 2023. 8

Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Martin Wattenberg, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in AlphaZero. *PNAS*, November 2022. 11, 23

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The Hydra Effect: Emergent Self-repair in Language Model Computations. *CoRR*, July 2023. 15, 16, 17, 23, 25

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and Editing Factual Associations in GPT. *NeurIPS*, 2022a. 13, 14, 15, 16, 17

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-Editing Memory in a Transformer. *ICLR*, 2022b. 16

William Merrill, Nikolaos Tsilivis, and Aman Shukla. A Tale of Two Circuits: Grokking as Competition of Sparse and Dense Subnetworks. *CoRR*, 2023. 19

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. A Mechanism for Solving Relational Tasks in Transformer Language Models. *CoRR*, May 2023. 26

Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The Quantization Model of Neural Scaling. *CoRR*, March 2023. 7, 19, 26

Eric J. Michaud, Isaac Liao, Vedang Lad, Ziming Liu, Anish Mudide, Chloe Loughridge, Zifan Carl Guo, Tara Rezaei Kheirkhah, Mateja Vukelić, and Max Tegmark. Opening the AI black box: program synthesis via mechanistic interpretability. *CoRR*, February 2024. 21, 24

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. *NeurIPS*, October 2013. 7

Joseph Miller and Clement Neo. We Found An Neuron in GPT-2. *AI Alignment Forum*, February 2023. 19

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, February 2019. 16

Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kıcıman, Hamid Palangi, Barun Patra, and Robert West. A Glitch in the Matrix? Locating and Detecting Language Model Grounding with Fakepedia. *CoRR*, 2023. 26

Basel Mousi, Nadir Durrani, and Fahim Dalvi. Can LLMs facilitate interpretation of pre-trained language models? *EMNLP*, 2023. 20

Jesse Mu and Jacob Andreas. Compositional Explanations of Neurons. *NeurIPS*, June 2020. 3, 4, 16

Jatin Nainani. Evaluating Brain-Inspired Modular Training in Automated Circuit Discovery for Mechanistic Interpretability. *CoRR*, January 2024. 20

Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L. Edelman, Fred Zhang, and Boaz Barak. SGD on Neural Networks Learns Functions of Increasing Complexity. *NeurIPS*, May 2019. 18

Neel Nanda. 200 COP in MI: Looking for Circuits in the Wild. *Neel Nanda's Blog*, 2022a. 26

Neel Nanda. 200 COP in MI: Analysing Training Dynamics. *Neel Nanda's Blog*, 2022b. 26

Neel Nanda. 200 COP in MI: Studying Learned Features in Language Models. *Neel Nanda's Blog*, 2022c. 26

Neel Nanda. A Comprehensive Mechanistic Interpretability Explainer & Glossary. *Neel Nanda's Blog*, December 2022d. 1

Neel Nanda. 200 COP in MI: Exploring Polysemanticity and Superposition. *Neel Nanda's Blog*, January 2023a. 25

Neel Nanda. 200 COP in MI: Techniques, Tooling and Automation. *Neel Nanda's Blog*, 2023b. 26

Neel Nanda. Actually, Othello-GPT Has A Linear Emergent World Representation. *Neel Nanda's Blog*, March 2023c. 11

Neel Nanda. Attribution Patching: Activation Patching At Industrial Scale. *Neel Nanda's Blog*, February 2023d. 15, 16

Neel Nanda. How to Think About Activation Patching. *AI Alignment Forum*, April 2023e. 15, 19

Neel Nanda. Mechanistic Interpretability Quickstart Guide. *Neel Nanda's Blog*, January 2023f. 1, 26

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *ICLR*, January 2023a. 7, 19, 26, 27

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent Linear Representations in World Models of Self-Supervised Sequence Models. *BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, September 2023b. 7, 9, 25

Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective. *CoRR*, December 2022. 10

Eshaan Nichani, Alex Damian, and Jason D. Lee. How Transformers Learn Causal Structure with Gradient Descent. *CoRR*, February 2024. 9

NicholasKees and janus. Searching for Search. *AI Alignment Forum*, November 2022. 10

nostalgebraist. interpreting GPT: the logit lens. *AI Alignment Forum*, August 2020. 12

Chris Olah. Distributed Representations: Composition & Superposition. *Transformer Circuits Thread*, 2023. 24

Chris Olah and Shan Carter. Research Debt. *Distill*, March 2017. 2

Chris Olah and Adam Jermyn. Reflections on Qualitative Research. *Transformer Circuits Thread*, March 2024. 16

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature Visualization. *Distill*, November 2017. 16

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The Building Blocks of Interpretability. *Distill*, March 2018. 1

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. *Distill*, March 2020. 1, 3, 4, 7, 8, 19

Christopher Olah. Mechanistic Interpretability, Variables, and the Importance of Interpretable Bases. *Transformer Circuits Thread*, 2022. 1, 3

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, December 1997. 12, 13

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context Learning and Induction Heads. *Transformer Circuits Thread*, 2022. 8, 18, 19, 26

Laura O'Mahony, Vincent Andrearczyk, Henning Muller, and Mara Graziani. Disentangling Neuron Representations with Concept Vectors. *CVPR Workshops*, April 2023. 7

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *CoRR*, 2022. 27

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C. Wallace, and David Bau. Future Lens: Anticipating Subsequent Tokens from a Single Hidden State. *CoNLL*, 2023. 12, 26

Vedant Palit, Rohan Pandey, Aryaman Arora, and P. Liang. Towards vision-language mechanistic interpretability: A causal tracing tool for BLIP. *ICCVW*, 2023. 27

Kiho Park, Yo Joong Choe, and Victor Veitch. The Linear Representation Hypothesis and the Geometry of Large Language Models. *NeurIPS Workshop on Causal Representation Learning*, November 2023. 25

Judea Pearl. *Causality*. Cambridge University Press, 2009. 9, 16

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets. *CoRR*, January 2022. 19

Philip Quirke and Fazl Barez. Understanding Addition in Transformers. *CoRR*, October 2023. 7, 26

Philip Quirke, Clement Neo, and Fazl Barez. Increasing Trust in Language Models through the Reuse of Verified Circuits. *CoRR*, February 2024. 26

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. *ACL*, July 2020. 17

Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. Defining and Quantifying the Emergence of Sparse Concepts in DNNs. *CoRR*, April 2023. 24

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *NAACL*, August 2016. 2, 17

___RicG___. AGI-Automated Interpretability is Suicide. *LessWrong*, May 2023. 26

Jonathan Richens and Tom Everitt. Robust agents learn causal world models. *ICLR Oral*, February 2024. 9

Ryan Riegel, Alexander Gray, Francois Luus, Naweed Khan, Ndivhuwo Makondo, Ismail Yunus Akhalwaya, Haifeng Qian, Ronald Fagin, Francisco Barahona, Udit Sharma, Shajith Ikbal, Hima Karanam, Sumit Neelam, Ankita Likhyani, and Santosh Srivastava. Logical Neural Networks. *NeurIPS*, June 2020. 17, 24

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, May 2019. 16

Thane Ruthenis. Internal Interfaces Are a High-Priority Interpretability Target. *AI Alignment Forum*, December 2022. 9

Thane Ruthenis. World-Model Interpretability Is All We Need. *AI Alignment Forum*, January 2023. 9

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks. *TMLR*, August 2023. 1, 11, 16, 22, 23, 24, 25

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. Neuron-level Interpretation of Deep NLP Models: A Survey. *TACL*, November 2022. 3

Mansi Sakarvadia, Aswathy Ajith, Arham Khan, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. Memory Injections: Correcting Multi-Hop Reasoning Failures during Inference in Transformer-Based Language Models. *CoRR*, September 2023a. 7

Mansi Sakarvadia, Arham Khan, Aswathy Ajith, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. Attention Lens: A Tool for Mechanistically Interpreting the Attention Head Information Retrieval Mechanism. *CoRR*, October 2023b. 12

Emmanuelle Salin, Badreddine Farah, Stéphane Ayache, and Benoit Favre. Are Vision-Language Transformers Learning Multimodal Representations? A Probing Perspective. *AAAI*, June 2022. 27

Tommaso Salvatori, Ankur Mali, Christopher L. Buckley, Thomas Lukasiewicz, Rajesh P. N. Rao, Karl Friston, and Alexander Ororbia. Brain-Inspired Computational Intelligence via Predictive Coding. *CoRR*, 2023. 9

Naomi Saphra. Interpretability Creationism. *The Gradient*, 2023. 19

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are Emergent Abilities of Large Language Models a Mirage? *CoRR*, May 2023. 18

Adam Scherlis, Kshitij Sachan, Adam S. Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and Capacity in Neural Networks. *CoRR*, July 2023. 5, 6, 23, 24

Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. *ICCV*, 2016. 2

Lloyd S. Shapley. A value for n -person games. *Cambridge University Press*, October 1988. 2

Lee Sharkey. Circumventing interpretability: How to defeat mind-readers. *CoRR*, December 2022. 23

Lee Sharkey. A technical note on bilinear layers for interpretability. *CoRR*, May 2023. 18, 22

Lee Sharkey, Sid Black, and beren. Current themes in mechanistic interpretability research. *AI Alignment Forum*, November 2022a. 1, 7

Lee Sharkey, Dan Braun, and Beren Millidge. Taking features out of superposition with sparse autoencoders. *AI Alignment Forum*, 2022b. 6, 12, 13

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards Understanding Sycophancy in Language Models. *CoRR*, October 2023. 24

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. *ICML*, 2017. 2

James B. Simon, Maksis Knutins, Liu Ziyin, Daniel Geisz, Abraham J. Fetterman, and Joshua Albrecht. On the Stepwise Nature of Self-Supervised Learning. *ICML*, May 2023. 18

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise. *CoRR*, June 2017. 2

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *JMLR*, 2014. 4

Dashiell Stander, Qinan Yu, Honglu Fan, and Stella Biderman. Grokking Group Multiplication with Cosets. *CoRR*, 2023. 19

Jacob Steinhardt. Emergent Deception and Emergent Optimization. *Bounded Regret*, February 2023. 18, 24

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. A Mechanistic Interpretation of Arithmetic Reasoning in Language Models using Causal Mediation Analysis. *EMNLP*, October 2023. 14, 19, 20

Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C. Love, Erin Grant, Iris Groen, Jascha Achterberg, Joshua B. Tenenbaum, Katherine M. Collins, Katherine L. Hermann, Kerem Oktar, Klaus Greff, Martin N. Hebart, Nori Jacoby, Qiuyi Zhang, Raja Marjieh, Robert Geirhos, Sherol Chen, Simon Kornblith, Sunayana Rane, Talia Konkle, Thomas P. O'Connell, Thomas Unterthiner, Andrew K. Lampinen, Klaus-Robert Müller, Mariya Toneva, and Thomas L. Griffiths. Getting aligned on representational alignment. *CoRR*, November 2023. 8

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. *ICML*, June 2017. 2

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution Patching Outperforms Automated Circuit Discovery. *CoRR*, October 2023. 15, 20

Max Tegmark and Steve Omohundro. Provably safe systems: the only path to controllable AGI. *CoRR*, September 2023. 26

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT Rediscovers the Classical NLP Pipeline. *ACL*, August 2019. 11

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The Slingshot Mechanism: An Empirical Study of Adaptive Optimizers and the Grokking Phenomenon. *CoRR*, 2022. 19

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear Representations of Sentiment in Large Language Models. *CoRR*, October 2023. 7

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function Vectors in Large Language Models. *CoRR*, 2023. 7

Dweep Trivedi, Jesse Zhang, Shao-Hua Sun, and Joseph J. Lim. Learning to Synthesize Programs as Interpretable and Generalizable Policies. *NeurIPS*, 2021. 24

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation Addition: Steering Language Models Without Optimization. *CoRR*, September 2023. 7

TurnTrout, peligrietzer, Ulisse Mini, montemac, and David Udell. Understanding and controlling a maze-solving policy network. *AI Alignment Forum*, November 2023. 27

Dmitry Vaintrob, jake_mendel, and Kaarel. Toward A Mathematical Framework for Computation in Superposition. *AI Alignment Forum*, 2024. 25, 26

Alexandre Variengien and Eric Winsor. Look Before You Leap: A Universal Emergent Decomposition of Retrieval Tasks in Language Models. *CoRR*, December 2023. 9, 20, 26

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *CoRR*, September 2023. 19

Abhinav Verma, Hoang M. Le, Yisong Yue, and Swarat Chaudhuri. Imitation-Projected Programmatic Reinforcement Learning. *NeurIPS*, 2019a. 24

Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically Interpretable Reinforcement Learning. *CoRR*, April 2019b. 24

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. *NeurIPS*, 2020. 13, 15, 17

M. Vilas, Timothy Schaumlöffel, and Gemma Roig. Analyzing vision transformers for image classification in class embedding space. *CoRR*, 2023. 27

Elena Voita and Ivan Titov. Information-Theoretic Probing with Minimum Description Length. *EMNLP*, March 2020. 11

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in Large Language Models: Dead, N-gram, Positional. *CoRR*, September 2023. 3

Johannes von Oswald, Eyvind Niklasson, Maximilian Schlegel, Seijin Kobayashi, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Blaise Agüera y Arcas, Max Vladymyrov, Razvan Pascanu, and João Sacramento. Uncovering mesa-optimization algorithms in Transformers. *CoRR*, September 2023. 23

Chelsea Voss, Gabriel Goh, Nick Cammarata, Michael Petrov, Ludwig Schubert, and Chris Olah. Branch Specialization. *Distill*, April 2021. 8, 19

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 small. *ICLR*, 2023. 8, 13, 14, 15, 19, 27

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 2020. 2

Sumio Watanabe. *Algebraic Geometry and Statistical Learning Theory*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2009. 18

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent Abilities of Large Language Models. *TMLR*, October 2022. 18

John Wentworth. How To Go From Interpretability To Alignment: Just Retarget The Search. *AI Alignment Forum*, August 2022. 9

James C. R. Whittington, Will Dorrell, Surya Ganguli, and Timothy E. J. Behrens. Disentangling with Biological Constraints: A Theory of Functional Cell Types. *CoRR*, September 2022. 12

Lionel Wong, Gabriel Grand, Alexander K. Lew, Noah D. Goodman, Vikash K. Mansinghka, Jacob Andreas, and Joshua B. Tenenbaum. From Word Models to World Models: Translating from Natural Language to the Probabilistic Language of Thought. *CoRR*, June 2023. 22

Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. BackdoorBench: A Comprehensive Benchmark of Backdoor Learning. *NeurIPS Datasets and Benchmarks*, October 2022. 25

Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D. Goodman. Interpretability at Scale: Identifying Causal Mechanisms in Alpaca. *CoRR*, May 2023. 17, 19, 21

Qinan Yu, Jack Merullo, and Ellie Pavlick. Characterizing Mechanisms for Factual Recall in Language Models. *CoRR*, October 2023. 8

Zeyu Yun, Yubei Chen, Bruno A. Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. *NAACL Workshop DeeLIO*, 2021. 12

Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. *ECCV*, 2014. 11

Biao Zhang, Ivan Titov, and Rico Sennrich. Sparse Attention with Linear Units. *EMNLP*, October 2021. 18

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, February 2017. 18

Fred Zhang and Neel Nanda. Towards Best Practices of Activation Patching in Language Models: Metrics and Methods. *ICLR*, 2023. 15

Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting CNNs via Decision Trees. *CVPR*, 2019. 24

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The Clock and the Pizza: Two Stories in Mechanistic Explanation of Neural Networks. *CoRR*, 2023. 19

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation Engineering: A Top-Down Approach to AI Transparency. *CoRR*, October 2023. 2, 7, 11, 20