Mechanistic Interpretability for AI Safety
A Review

Anonymous authors
Paper under double-blind review

Abstract

Understanding AI systems' inner workings is critical for ensuring value alignment and safety. This review explores mechanistic interpretability: reverse engineering the computational mechanisms and representations learned by neural networks into human-understandable algorithms and concepts to provide a granular, causal understanding. We establish foundational concepts such as features encoding knowledge within neural activations and hypotheses about 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, along with risks such as capability gains and dual-use concerns. We investigate challenges surrounding scalability, automation, and comprehensive interpretation. We advocate for clarifying concepts, setting standards, and scaling techniques to handle complex models and behaviors and expand to domains such as vision and reinforcement learning. Mechanistic interpretability could help prevent catastrophic outcomes as AI systems become more powerful and inscrutable.

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 trustworthy and beneficial.

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

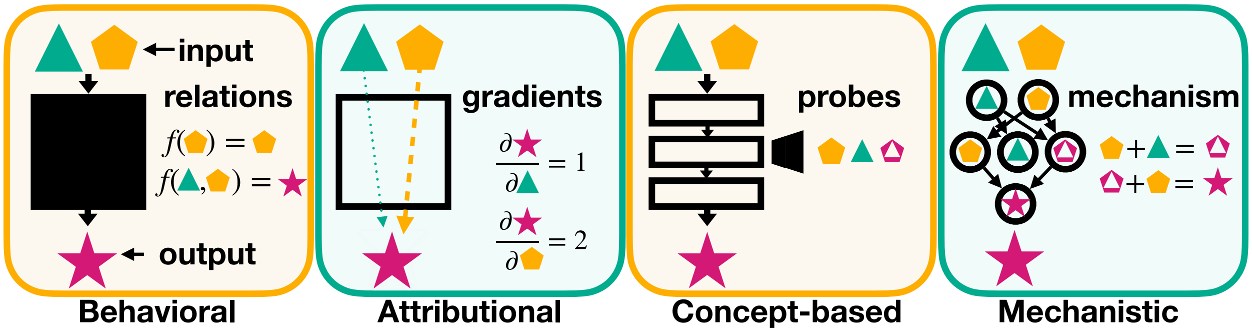


Figure 1: Interpretability paradigms offer distinct lenses for understanding neural networks: **Behavioral** analyzes input-output relations; **Attributional** quantifies individual input feature influences; **Concept-based** identifies high-level representations governing behavior; **Mechanistic** uncovers 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), presenting a taxonomy and survey of the current field (Section 5), exploring relevance to AI safety (Section 6), and addressing challenges (Section 7) and future directions (Section 8).

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 include training supervised auxiliary classifiers (Belinkov, 2021), employing unsupervised contrastive and structured probes (see Section 4.1) to explore latent knowledge (Burns et al., 2023), and using neural representation analysis to quantify the representational similarities between the internal representations learned by different neural networks (Kornblith et al., 2019; Bansal et al., 2021). Beyond observational analysis, concept-based interpretability can enable manipulation of these representations – also called *rep-*

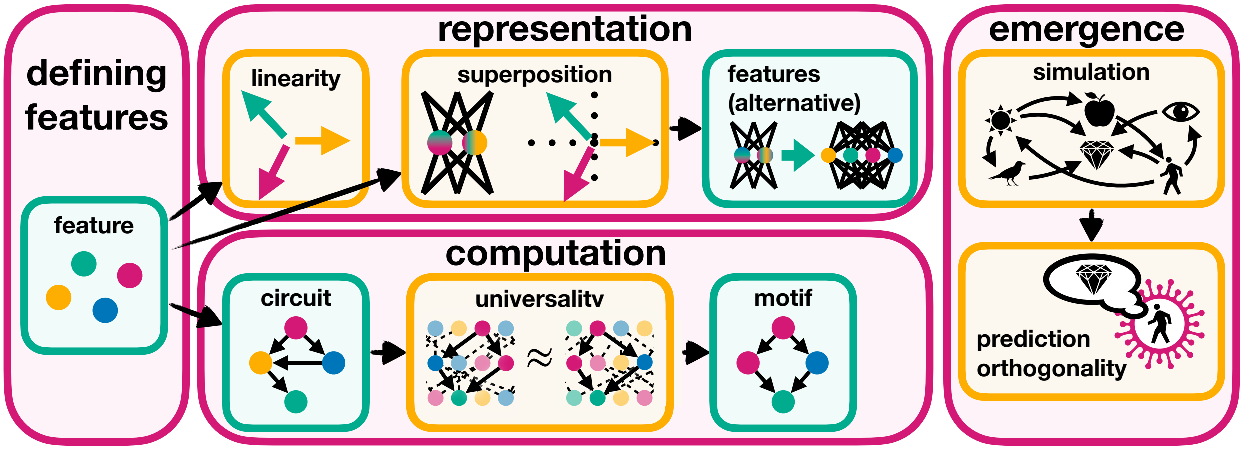


Figure 2: Overview of key concepts and hypotheses in mechanistic interpretability, organized into four subsection (pink boxes): defining features (Section 3.1), representation (Section 3.2), computation (Section 3.3), and emergence (Section 3.4). In turquoise, it highlights definitions like *features*, *circuits*, and *motifs*, and in orange, it highlights hypotheses like *linear representation*, *superposition*, *universality*, *simulation*, and *prediction orthogonality*. Arrows show relationships, e.g., superposition enabling an alternative feature definition or universality connecting circuits and motifs.

resentation 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 key concepts and hypotheses of mechanistic interpretability, as summarized in Figure 2. We start by defining features as the basic units of representation (Section 3.1). We then examine the nature of these features, including the challenges posed by polysemantic neurons and the implications of the superposition and linear representation hypotheses (Section 3.2). Next, we explore computation through circuits and motifs, considering the universality hypothesis (Section 3.3). Finally, we discuss the implications for understanding emergent properties, such as internal world models and simulated agents with potentially misaligned objectives (Section 3.4).

3.1 Defining Features as Representational Primitives

Features as fundamental units of representation. The notion of a *feature* in neural networks is central yet elusive, reflecting the pre-paradigmatic state of mechanistic interpretability. We adopt the notion of *features* as the *fundamental units of neural network representations*, such that features cannot be further *disentangled* into simpler, distinct factors. These features are core components of a neural network's representation, analogous to how cells form the fundamental unit of biological organisms (Olah et al., 2020).

Concepts as natural abstractions. The world consists of various entities that can be grouped into categories or *concepts* based on shared properties. These concepts form high-level summaries like "tree" or

Definition 1: Feature

Features are the fundamental units of neural network representations that cannot be further decomposed into simpler independent factors.

"velocity," allowing compact world representations by discarding many irrelevant low-level details. Neural networks can capture and represent such *natural abstractions* (Chan et al., 2023) through their learned *features*, which serve as building blocks of their internal representations, aiming to capture the *concepts* underlying the data.

Features encoding input patterns. In traditional machine learning, *features* are understood as characteristics or attributes derived directly from the input data stream (Bishop, 2006). This view is particularly relevant for systems focused on *perception*, where features map closely to the input data. However, in more advanced systems capable of *reasoning* with abstractions, features may emerge internally within the model as representational patterns, even when processing information unrelated to the input. In this context, features are better conceptualized as *any measurable property or characteristic of a phenomenon* (Olah, 2022), encoding abstract concepts rather than strictly reflecting input attributes.

Features as representational atoms. A key property of features is their irreducibility, meaning they cannot be decomposed into or expressed as a combination of simpler, independent factors. In the context of input-related features, Engels et al. (2024) define a feature as *irreducible* if it cannot be decomposed into or expressed as a combination of statistically independent patterns or factors in the original input data. Specifically, a feature is reducible if transformations reveal its underlying pattern, which can be separated into independent co-occurring patterns or is a mixture of patterns that never co-occur. We propose generalizing this notion of irreducibility to features encoding abstract concepts not directly tied to input patterns, such that features cannot be reduced to combinations or mixtures of other independent components within the model's representations.

Features beyond human interpretability. Features could be defined from a *human-centric perspective* as *semantically meaningful, articulable input patterns encoded in the network's activation space* (Olah, 2022). However, while cognitive systems may converge on similar *natural abstractions* (Chan et al., 2023), these need not necessarily align with human-interpretable *concepts*. Adversarial examples have been interpreted as non-interpretable features meaningful to models but not humans. Imperceptible perturbations fool networks, suggesting reliance on alien representational patterns (Ilyas et al., 2019). As models surpass human capabilities, their learned features may become increasingly abstract, encoding information in ways incongruent with human intuition (Hubinger, 2019a). Mechanistic interpretability aims to uncover the *actual* representations learned, even if diverging from human concepts. While human-interpretable concepts provide guidance, a non-human-centric perspective that defines features as independent model components, whether aligned with human concepts or not, is a more comprehensive and future-proof approach.

3.2 Nature of Features: From Monosemantic Neurons to Non-Linear 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, as shown in Figure 3. 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).

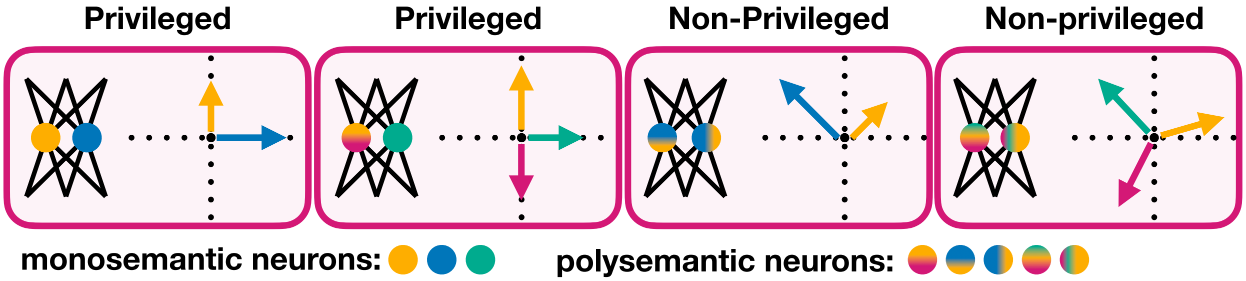


Figure 3: Contrasting privileged and non-privileged bases. In a non-privileged basis, there is no reason to expect features to be basis-aligned – calling basis dimensions neurons has no meaning. In a privileged basis, the architecture treats basis directions differently – features can but need not align with neurons (Bricken et al., 2023). **Leftmost:** Privileged basis; individual features (arrows) align with basis directions, resulting in *monosemantic* neurons (colored circles). **Middle left:** Privileged basis, where despite having more features than neurons, some neurons are monosemantic, representing individual features, while others are *polysemantic* (overlapping gradients), encoding *superposition* of multiple features. **Middle right:** Non-privileged basis where, even when the number of features equals the number of neurons, the lack of alignment between the feature directions and basis directions results in polysemantic neurons encoding combinations of features. **Rightmost:** Non-privileged, polysemantic neurons as feature directions do not align with neuron basis.

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 representational primitives of neural networks, 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 representational 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 3, middle right).
- 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 1: 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 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 5). Consequently, an alternative definition of features could be:

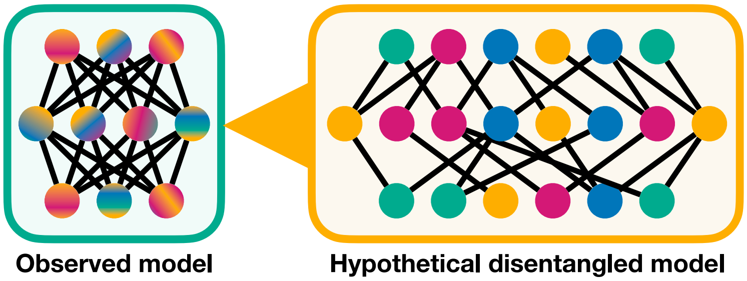


Figure 5: Observed neural networks (left) can be viewed as compressed simulations of larger, sparser networks (right) where neurons represent distinct features. An "almost orthogonal" projection compresses the high-dimensional sparse representation, manifesting as polysemantic neurons involved with multiple 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

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 $\mathbf{x}'$:

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

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.

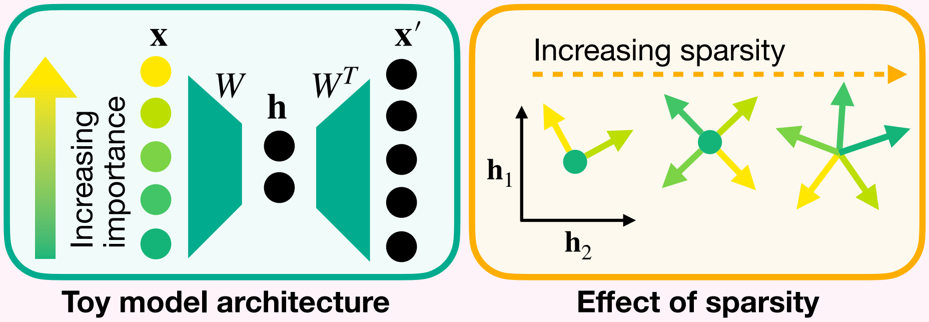


Figure 4: Illustration of the toy model architecture and the effects of sparsity. (left) Transformation of a five-feature input vector $\mathbf{x}$ into a two-dimensional hidden representation $\mathbf{h}$, and its reconstruction as $\mathbf{x}'$ using the weight matrix W and its transpose, with feature 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.

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

Definition 2: 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.

enhance our comprehension of neural networks by providing a more interpretable and manipulable framework for their internal representations.

Hypothesis 2: Linear Representation

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

The *linear representation* hypothesis suggests that neural networks frequently represent high-level features as linear directions in activation space. This hypothesis can simplify the understanding and manipulation of neural network representations (Nanda et al., 2023b). The prevalence of linear layers in neural network architectures favors linear representations. Matrix multiplication in these layers most readily processes linear features, while more complex non-linear encodings would require multiple layers to decode.

However, recent work by Engels et al. (2024) provides evidence against a strict formulation of the linear representation hypothesis by identifying circular features representing days of the week and months of the year. These multi-dimensional, non-linear representations were shown to be used for solving modular arithmetic problems in days and months. Intervention experiments confirmed that these circular features are the fundamental unit of computation in these tasks, and the authors developed methods to decompose the hidden states, revealing the circular representations.

Establishing non-linearity can be challenging. For example, Li et al. (2023a) initially found that in a GPT model trained on Othello, the board state could only be decoded with a non-linear probe when represented in terms of "black" and "white" pieces, seemingly violating the linearity assumption. However, Nanda (2023c); Nanda et al. (2023b) later showed that a linear probe sufficed when the board state was decoded in terms of "own" and "opponent's" pieces, reaffirming the linear representation hypothesis in this case. In contrast, the work by Engels et al. (2024) provides a clear and convincing existence proof for non-linear, multi-dimensional representations in language models.

Despite these exceptions, empirical evidence largely supports the linear representation hypothesis in many contexts, especially for feedforward networks with ReLU activations. Semantic vector calculus in word embeddings (Mikolov et al., 2013), successful linear probing (Alain & Bengio, 2016; Belinkov, 2021), sparse dictionary learning (Bricken et al., 2023; Cunningham et al., 2024; Deng et al., 2023), and linear decoding of 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) in large language models all point to the prevalence of linear representations. Moreover, linear addition techniques 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.

While the linear representation hypothesis remains a useful simplification, it is important to recognize its limitations and the potential role of non-linear representations (Sharkey et al., 2022a). As neural networks continue to evolve, ongoing reevaluation of the hypothesis is crucial, particularly considering the possible emergence of non-linear features under optimization pressure for interpretability (Hubinger, 2022). Alternative perspectives, such as the polytope lens proposed by Black et al. (2022), emphasize the impact of non-linear activation functions and discrete polytopes formed by piecewise linear activations as potential primitives of neural network representations.

3.3 Circuits as Computational Primitives and Motifs as Universal Circuit Patterns

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).

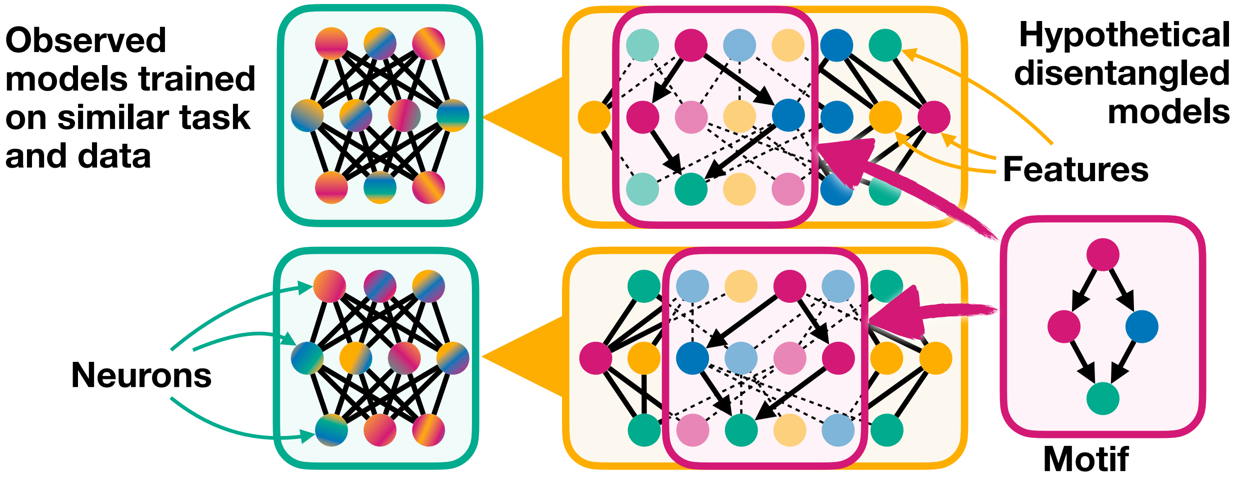

Observed models trained on similar task and data
Neurons
Hypothetical disentangled models
Features
Motif

Figure 6: Comparing observed models (left) and corresponding hypothetical *disentangled* models (right) trained on similar tasks and data. The observed models show different neuronal activation patterns, while the dissection into feature-level *circuits* reveals a *motif* - a shared circuit pattern emerging across models, hinting at *universality* – models converging on similar solutions based on common underlying principles.

Definition 3: Circuit

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

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). Scaling such a *comprehensive* circuit analysis to broader behaviors in large language models remains challenging. However, there has been notable progress in scaling circuit analysis of *narrow behaviors* to larger circuits and models, such as indirect object identification (Wang et al., 2023) and greater-than computations (Hanna et al., 2023) in GPT-2 and multiple-choice question answering in Chinchilla (Lieberum et al., 2023).

In search of general and universal circuits, researchers focus particularly on more general and transferable behaviors. McDougall et al. (2023)'s work on copy suppression in GPT-2's attention heads sheds light on model calibration and self-repair mechanisms. Davies et al. (2023) and Feng & Steinhardt (2023) focus on how large language models represent symbolic knowledge through variable binding and entity-attribute binding, respectively. Yu et al. (2023) explore mechanisms for factual recall, 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.

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. Figure 6 contrasts observed neural network models with hypothetical disentangled models, illustrating how a shared circuit pattern can emerge across different models trained on similar tasks and data, hinting at an underlying *universality*.

Definition 4: 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, we can propose two versions for a *universality* hypothesis regarding the convergence of features and circuits across neural network models:

Hypothesis 3: Weak Universality

There are underlying principles governing how neural networks learn to solve certain tasks. Models will generally converge on analogous solutions that adhere to the common underlying principles. However, the specific *features* and *circuits* that implement these principles can vary across different models based on factors like hyperparameters, random seeds, and architectural choices.

Hypothesis 4: Strong Universality

The *same* core features and circuits will universally and consistently arise across all neural network models trained on similar tasks and data distributions and using similar techniques, reflecting a set of fundamental computational *motifs* that neural networks inherently gravitate towards when learning.

The universality hypothesis posits a convergence in forming features and circuits across various models and tasks, which could significantly ease interpretability efforts in AI. It 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.

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). Marchetti et al. (2023) provide mathematical support for emerging universal features.

While various studies support the universality hypothesis, questions remain about the extent of feature and circuit similarity across different models and tasks. 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), with evidence for the weak but not the strong formulation (Chughtai et al., 2023).

3.4 Emergence of World Models and Simulated Agents

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 *internal world models* that potentially form *implicitly* as a by-product of the training process, especially in LLMs trained on next-token prediction – also called GPT.

LLMs are sometimes characterized as *stochastic parrots* (Bender et al., 2021). This label stems from their fundamental operational mechanism of predicting the next word in a sequence, which is seen as relying heavily on memorization. From this viewpoint, LLMs are thought to form complex correlations based on observational data but lack the ability to develop causal models of the world due to their lack of access to interventional data (Pearl, 2009).

An alternative perspective on LLMs comes from 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.

The *simulation* hypothesis 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 5: 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, 2024), 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. Li et al. (2021) identified contextual word representations that function as models of entities and situations evolving throughout a discourse, akin to linguistic models of dynamic semantics. Patel & Pavlick (2022) demonstrated that LLMs can map conceptual domains (e.g., direction, color) to grounded world representations given a few examples, suggesting they learn rich conceptual spaces (Gardenfors, 2004) reflective of the non-linguistic world.

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 are simulators, simulating entities known as *simulacra* that can be either agentic or non-agentic, with different objectives from the simulator itself (janus, 2022). The implications of the simulation and prediction orthogonality hypotheses for AI safety and alignment are discussed in Section 6.

Hypothesis 6: Prediction Orthogonality

A model whose objective is prediction can simulate agents who optimize toward any objectives with any degree of optimality (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 7 provides an overview of the relevant methods and techniques.

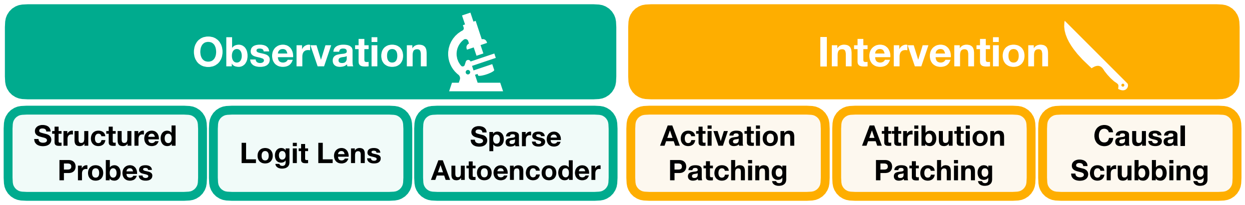


Figure 7: Overview of relevant methods and techniques employed in mechanistic interpretability research. Observational methods proposed for mechanistic interpretability include structured probes (more aligned with top-down interpretability), logit lens variants, and sparse autoencoders (SAEs). Additionally, 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).

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 focusing on bottom-up, mechanistic interpretability approaches, we can also consider integrating top-down, concept-based structured probes with mechanistic interpretability.

Structured probes aid conceptual interpretability, probing language models for complex features like truth representations. Notably, Burns et al. (2023)'s contrast-consistent search (CCS) method identifies linear projections exhibiting logical consistency in hidden states, contrasting truth values for statements and negations.

However, structured probes face significant challenges in unsupervised probing scenarios. As Farquhar et al. (2023) showed, arbitrary features, not just knowledge-related ones, can satisfy the CCS loss equally well, raising doubts about scalability. For example, the loss may capture *simulation* of knowledge from hypothesized *simulacra* within sufficiently powerful language models rather than the models' true knowledge. Furthermore, Farquhar et al. (2023) demonstrates unsupervised methods like CCS often detect prominent but unintended features in the data, such as distractors like "banana." The discovered features are also highly sensitive to prompt choice, and there is no principled way to select prompts that would reliably surface a model's true knowledge.

While structured probes primarily focus on high-level conceptual representations (Zou et al., 2023), 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 such as 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 classification layer (which projects the residual stream activation into logits/vocabulary space) to intermediate activations of the residual stream, revealing how prediction confidence evolves across computational stages. This is possible because transformers tend to build their predictions across layers iteratively (Geva et al., 2022). 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.

Sparse Dictionary Learning

Sparse autoencoders (Cunningham et al., 2024) represent a solution attempt to the challenge of *polysemantic* neurons. The problem of *superposition* is mathematically formalized as *sparse dictionary learning* (Olshausen & Field, 1997) problem to decompose neural network activations into *disentangled* 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 as sparse linear combinations. If successful, the learned dictionary contains *monosemantic* neurons corresponding to *features* (Sharkey et al., 2022b). The autoencoder architecture consists of an encoder and a ReLU activation function, expanding the input dimensionality to $d_{\text{hid}} > d_{\text{in}}$. 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.

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

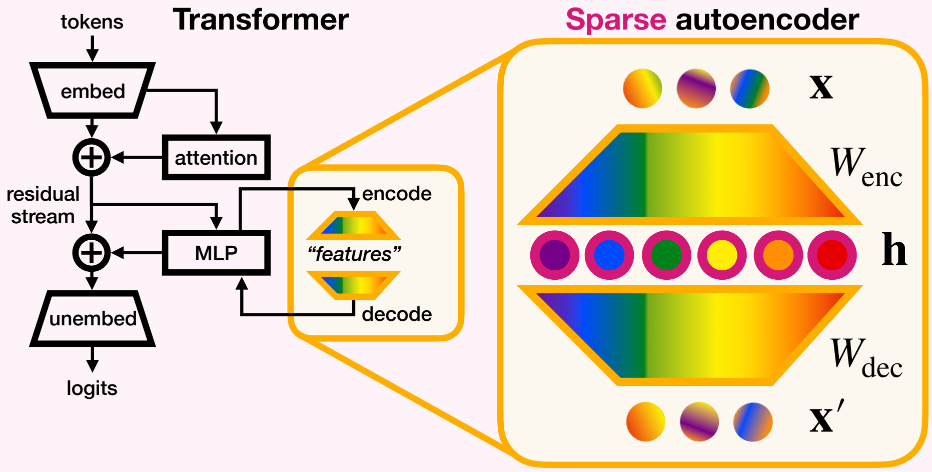


Figure 8: Illustration of a sparse autoencoder applied to the MLP layer activations, consisting of an encoder that increases dimensionality while emphasizing sparse representations and a decoder that reconstructs the original activations using the learned feature dictionary.

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). The "ground truth features" here refer to the true, disentangled features that underlie the data distribution, which the autoencoder aims to recover through its sparse encoding. In the context of neural networks, these would correspond to the individual features combined to form neuron activations, which the sparse autoencoder attempts to disentangle and represent explicitly in its dictionary.

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. We show an overview in the pink box on sparse autoencoders in Figure 8.

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., 2023a), making them an actively researched area in mechanistic interpretability.

4.2 Intervention

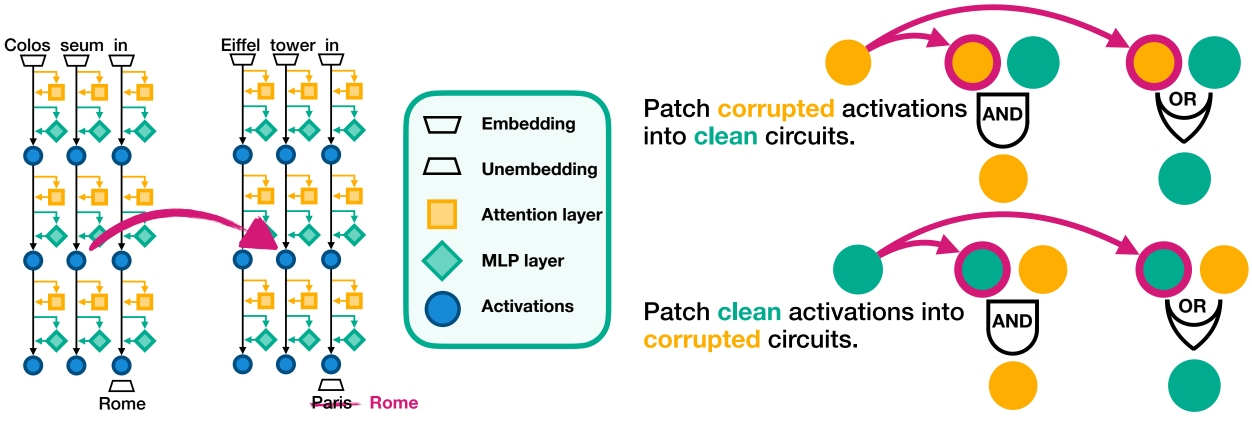
Causality as a Theoretical Foundation. The theory of causality (Pearl, 2009) provides a mathematically precise framework for mechanistic interpretability, offering a rigorous approach to understanding high-level semantics in neural representations (Geiger et al., 2023a). By treating neural networks as causal models, with their *compute graphs serving as causal graphs*, researchers can perform precise interventions and examine the roles of individual parameters (McGrath et al., 2023). This causal perspective on interpretability has led to the development of various intervention techniques, including activation patching (Section 4.2.1), causal abstraction (Section 4.2.2), and hypothesis testing methods (Section 4.2.3).

4.2.1 Activation Patching

Activation patching is a collective term for a set of causal intervention techniques that manipulate neural network activations to shed light on the decision-making processes within the model. These techniques, including 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), share the common goal of modifying 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 (Figure 9a).

The primary objective of activation patching 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. This enables researchers to infer the function and importance of those components. Key applications include localizing behavior by identifying critical activations, such as understanding the storage and processing of factual information (Meng et al., 2022a; Geva et al., 2023; Goldowsky-Dill et al., 2023; Stolfo et al., 2023), and analyzing component interactions through 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).

The standard protocol for activation patching (Figure 9a) involves: (step i) running the model with a clean input and caching the latent activations; (step ii) executing the model with a corrupted input; (step iii) rerunning the model with the corrupted input but substituting specific activations with those from the clean cache; and (step iv) determining significance by observing the variations in the model's output during the third step, thereby highlighting the importance of the replaced components. This process relies on comparing pairs of inputs: a clean input, which triggers the desired behavior, and a corrupted input, which is identical



(a) (b)

Figure 9: (a) Activation patching in a transformer model. Left: The model processes the clean input "Colosseum in Rome," caching the latent activations (step i). Right: The model runs with the corrupted input "Eiffel Tower in Paris" (step ii). The pink arrow shows an MLP layer activation (green diamond) patched from the clean run into the corrupted run (step iii). This causes the prediction to change from "Paris" to "Rome," demonstrating how the significance of the patched component is determined (step iv). By comparing these carefully selected inputs, researchers can control for confounding circuitry and isolate the specific circuit responsible for the location prediction behavior. (b) Activation patching directions: Top: Patching corrupted activations (orange) into clean circuits (turquoise) reveals *sufficient* components for identifying OR logic scenarios. Bottom: Patching clean activations (green) into corrupted circuits (orange) reveals *necessary* components that are useful for identifying AND logic scenarios. The AND and OR gates demonstrate how these patching directions uncover different logical relationships between model components.

to the clean one except for critical differences that prevent the behavior. By carefully selecting these inputs, researchers can *control for confounding circuitry* and isolate the specific circuit responsible for the behavior.

Differences in patching direction—clean to corrupted (causal tracing) versus corrupted to clean (resample ablation)—provide insights into the sufficiency or necessity of model components for a given behavior. Clean to corrupted patching identifies activations sufficient for restoring clean performance, even if they are unnecessary due to redundancy, which is particularly informative in OR logic scenarios (Figure 9b, A-OR-B). Conversely, corrupted to clean patching determines the necessary activations for clean performance, which is useful in AND logic scenarios (Figure 9b, A-AND-B).

Activation patching can employ corruption methods, including zero-, mean-, random-, or resample ablation, each modulating the model's internal state in distinct ways. Resample ablation stands out for its effectiveness in maintaining consistent model behavior by not changing the data distribution too much (Zhang & Nanda, 2023). However, it is essential to be careful when interpreting the patching results, as breaking behavior by taking the model off-distribution is uninteresting for finding the relevant circuit (Nanda, 2023e).

Path Patching and Subspace Activation Patching. Path patching extends the activation patching approach to multiple edges in the computational graph (Wang et al., 2023; Goldowsky-Dill et al., 2023), allowing for a more fine-grained analysis of component interactions (Figure 9c). For example, path patching can be used to estimate the direct and indirect effects of attention heads on the output logits. Subspace activation patching, also known as distributed interchange interventions (Geiger et al., 2023b), aims to intervene only on linear subspaces of the representation space where *features* are hypothesized to be encoded, providing a tool for more targeted interventions.

Limitations and Advancements. Activation patching has several limitations, including the effort required to design input templates and counterfactual datasets, the need for human inspection to isolate important subgraphs, and potential second-order effects that can complicate the interpretation of results

(Lange et al., 2023) and the *hydra effect* (McGrath et al., 2023) (see discussion in Section 7.2). Recent advancements aim to address these limitations, such as automated circuit discovery algorithms (Conmy et al., 2023), gradient-based methods for scalable component importance estimation like attribution patching (Nanda, 2023d; Syed et al., 2023), and techniques to mitigate self-repair interferences during analysis (Ferrando & Voita, 2024).

4.2.2 Causal Abstraction

Causal abstraction (Geiger et al., 2021a; 2023a) provides a mathematical framework for mechanistic interpretability, treating neural networks and their explanations as causal models. This approach validates explanations through interchange interventions on network activations (Jenner et al., 2023), unifying 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 (Geiger et al., 2023a).

To overcome computational limitations, Distributed Alignment Search (DAS) (Geiger et al., 2023b) introduced gradient-based distributed interchange interventions, extending causal abstraction to larger models like Alpaca (Wu et al., 2023b). Further advancements include Causal Proxy Models (CPMs) (Wu et al., 2023a), which address the challenge of counterfactual observations.

Applications of causal abstraction span from linguistic phenomena analysis (Arora et al., 2024; Wu et al., 2022b), and evaluation of interpretability methods (Huang et al., 2024), to improving performance through representation finetuning (Wu et al., 2024), and improving efficiency via model distillation (Wu et al., 2022b).

4.2.3 Hypothesis Testing

In addition to the causal abstraction framework, several methods have been developed for rigorous hypothesis testing about neural network behavior. These methods aim to formalize and empirically validate explanations of how neural networks implement specific behaviors.

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. This method replaces activations in G with others that should be equivalent according to the hypothesis, measuring performance on the scrubbed model to validate the hypothesis.

Locally consistent abstractions (Jenner et al., 2023) offer a more permissive approach, checking the consistency between the neural network and the explanation only one step away from the intervention node. This method forms a middle ground between the strictness of full causal abstraction and the flexibility of causal scrubbing.

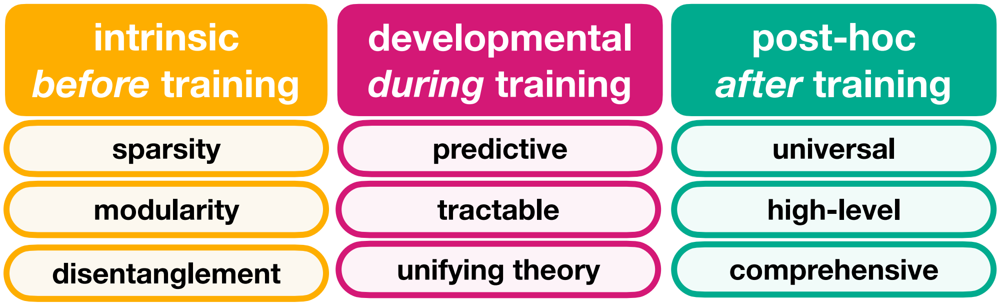
These methods form a hierarchy of strictness, with full causal abstractions being the most stringent, followed by locally consistent abstractions and causal scrubbing being the most permissive. This hierarchy highlights trade-offs in choosing stricter or more permissive notions, affecting the ability to find acceptable explanations, generalization, and mechanistic anomaly detection.

4.3 Integrating Observation and Intervention.

To comprehensively understand internal neural network mechanisms, combining observational and interventional methods is crucial. For instance, sparse autoencoders can be used to disentangle superposed features (Cunningham et al., 2024), followed by targeted activation patching to test the causal importance of these features (Wang et al., 2023). Similarly, the logit lens can track prediction formation across layers (nostalgebraist, 2020), with subsequent interventions confirming causal relationships at key points. Probing techniques can identify encoded information (Belinkov, 2021), which can then be subjected to causal abstraction (Geiger et al., 2023a) to understand how this information is utilized. This iterative refinement process, where broad observational methods guide targeted interventions and intervention results inform further observations, enables a multi-level analysis that builds a holistic understanding across different levels of abstraction. Recent work by Marks et al. (2024) and Braun et al. (2024) demonstrates the potential of integrating sparse autoencoders with automated circuits discovery (Conmy et al., 2023; Syed et al., 2023), combining feature-level analysis with circuit-level interventions to uncover the interplay between representation and mechanism. By systematically combining these complementary methods, researchers can generate and rigorously test hypotheses about neural network behavior, addressing challenges such as feature *superposition*.

5 Current Research

This section surveys current research in mechanistic interpretability across three approaches based on when and how the model is interpreted during training: Intrinsic interpretability methods are applied before training to enhance the model's inherent interpretability (Section 5.1). Developmental interpretability involves studying the model's learning dynamics and the emergence of internal structures during training (Section 5.2). After training, post-hoc interpretability techniques are applied to gain insights into the model's behavior and decision-making processes (Section 5.3), 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 5.4).



intrinsic *before* training	developmental *during* training	post-hoc *after* training
sparsity	predictive	universal
modularity	tractable	high-level
disentanglement	unifying theory	comprehensive

Figure 10: 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.

5.1 Intrinsic Interpretability

Intrinsic methods for mechanistic interpretability offer a promising approach to designing neural networks that are 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 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.

5.2 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.

5.3 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.

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., 2023b). 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., 2023b).

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 work that analyzes transformers trained on modular addition (Nanda et al., 2023a) 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. Techniques for *representation engineering* (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.

5.4 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.

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*.

6 Relevance

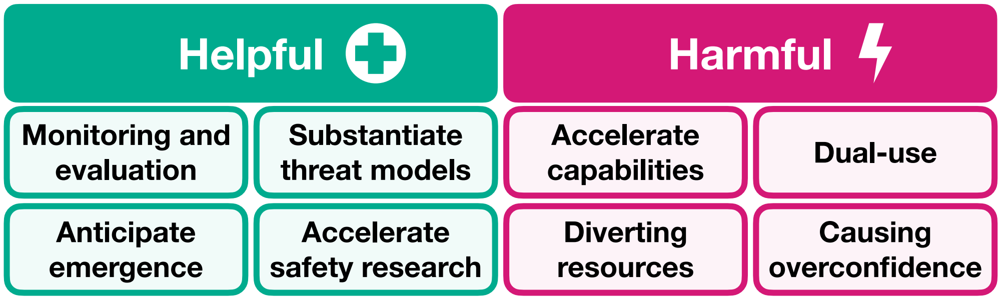


Figure 11: Potential benefits and risks of mechanistic interpretability for AI safety.

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 (Nanda, 2022e). 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 (Casper, 2023). It may also help anticipate emergent capabilities, such as the emergence of new skills or behaviors in the model before they fully manifest (Wei et al., 2022; Steinhardt, 2023; Nanda et al., 2023a; Barak et al., 2022). This relates to studying the incremental development of internal structures and representations as the model learns (Section 5.2). 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) (Hubinger et al., 2019; von Oswald et al., 2023). It may also trigger normative shifts within the AI community toward rigorous safety protocols by revealing potential risks or concerning behaviors (Hubinger, 2019a).

Regarding specific AI risks, interpretability may prevent malicious misuse by locating and erasing sensitive information stored in the model (Meng et al., 2022a; Nguyen et al., 2022). 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 (Hendrycks & Mazeika, 2022). Interpretability can provide safety filters for every stage of training: before training by deliberate design (Hubinger, 2019a), during training by detecting early signs of misalignment and potentially shifting the distribution towards alignment (Hubinger, 2022; Sharkey, 2022), and after training by rigorous evaluation of artificial cognition for honesty (Burns et al., 2023; Zou et al., 2023) and screening for deceptive behaviors (Park et al., 2023b).

The emergence of *internal world models* in LLMs, as posited by the *simulation* hypothesis, could have significant implications for AI alignment research. Finding an internal representation of human values and aiming the AI system's objective may be a trivial way to achieve alignment (Wentworth, 2022), especially if the world model is internally separated from notions of goals and agency (Ruthenis, 2022). In such cases, world model interpretability alone may be sufficient for alignment (Ruthenis, 2023).

Conditioning pre-trained models is considered a comparatively safe pathway towards general intelligence, as it avoids directly creating agents with inherent goals or agendas (Jozdien, 2022; Hubinger et al., 2023). However, 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 (Hubinger et al., 2023). In contrast, reinforcement learning tends to create agents by default (Casper et al., 2023a; Ngo et al., 2022).

The *prediction orthogonality* hypothesis suggests that prediction-focused models like GPT can simulate agents with potentially misaligned objectives (janus, 2022). Although GPT may lack genuine agency or intentionality, it may produce outputs that simulate these qualities (Bereska & Gavves, 2023). This underscores the need for careful *oversight* and, better yet, using mechanistic interpretability to search for internal agents or their constituents, such as optimization or search processes – an endeavor known as *searching for search* (NicholasKees & janus, 2022).

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 (technicalities & Stag, 2023; Hubinger, 2020). It could enhance strategies like detecting *deceptive alignment* (when a model appears aligned but is actually pursuing different goals) (Park et al., 2023b), *eliciting latent knowledge* from models (Christiano et al., 2021), and enabling better scalable *oversight*, such as in *iterative distillation and amplification* (Chan, 2023). A high degree of understanding may even allow for *well-founded AI* approaches (AI systems with provable guarantees) (Tegmark & Omohundro, 2023) or *microscope AI* (extract world knowledge from the model without letting the model take actions) (Hubinger, 2019a). 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 (Wentworth, 2022). 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 (Soares, 2023; Kross, 2023; Hendrycks & Mazeika, 2022). While historically, interpretability research had little impact on AI capabilities, recent exceptions like discoveries about scaling laws (Hoffmann et al., 2022), architectural improvements inspired by studying induction heads

(Olsson et al., 2022; Fu et al., 2023a; Poli et al., 2023; Schuster et al., 2022), and efficiency gains inspired by the logit lens technique (Schuster et al., 2022) demonstrated its potential impact. Scaling interpretability research may necessitate automation (Conmy et al., 2023; Bills et al., 2023), potentially enabling rapid self-improvement of AI systems (___RicG___, 2023). Some researchers recommend selective publication and focusing on lower-risk areas to mitigate these risks (Hobbhahn & Chan, 2023; Shovelain & McKernon, 2023; Elhage et al., 2022b; Nanda et al., 2023a).

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) (Nguyen et al., 2022; Pochinkov & , 2023) but could be misused for censorship. Similarly, while interpretability may help improve adversarial robustness (Räuker et al., 2023), it may also facilitate the development of stronger adversarial attacks (Mu & Andreas, 2020; Casper et al., 2023b).

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 (Charbel-Raphaël, 2023; Casper, 2023). Robust evaluation and benchmarking (Section 8.2) are crucial to validate interpretability claims and reduce the risks of overinterpretation or misinterpretation.

7 Challenges

7.1 Research Issues

Need for Comprehensive, Multi-Pronged Approaches. Current interpretability research often focuses on individual techniques rather than combining complementary approaches. To achieve a holistic understanding of neural networks, we propose utilizing a diverse interpretability toolbox that integrates multiple methods (see also Section 4.3), such as: *i.)* Coordinating observational (e.g., probing, logit lens) and interventional methods (e.g., activation patching) to establish causal relationships. *ii.)* Combining feature-level analysis (e.g., sparse autoencoders) with circuit-level interventions (e.g., path patching) to uncover representation-mechanism interplay. *iii.)* Integrating intrinsic interpretability approaches with post-hoc analysis for robust understanding.

For example, coordinated methods could be used for *reverse engineering* trojaned behaviors (Casper et al., 2023c), where observational techniques identify suspicious activations, interventional methods isolate the relevant circuits, and intrinsic approaches guide the design of more robust architectures.

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.

7.2 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., 2023b; 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.

8 Future Directions

Given the current limitations and challenges, several key research problems emerge as critical for advancing mechanistic interpretability. These problems span four main areas: emphasizing conceptual clarity Section 8.1, establishing rigorous standards Section 8.2, improving the scalability of interpretability techniques Section 8.3, and expanding the research scope Section 8.4. Each subsection presents specific research questions and challenges that need to be addressed to move the field forward.

8.1 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

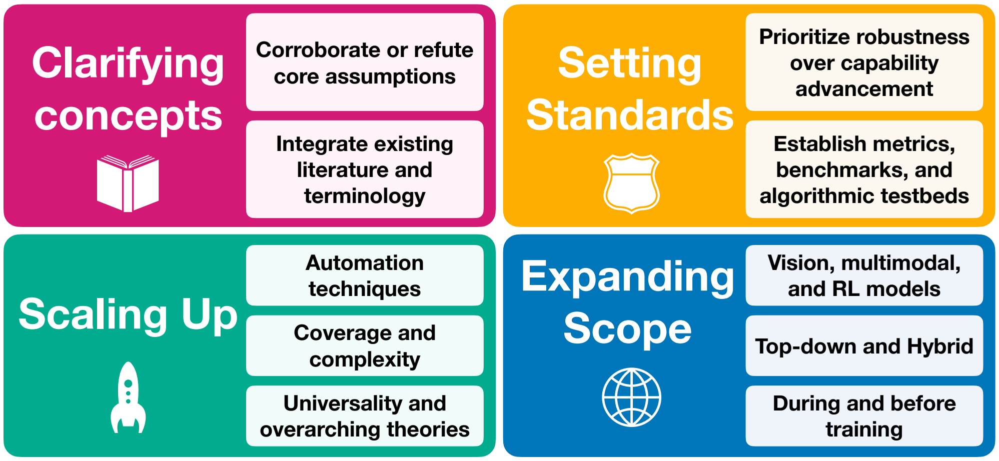

Clarifying concepts
Corroborate or refute core assumptions
Integrate existing literature and terminology
Setting Standards
Prioritize robustness over capability advancement
Establish metrics, benchmarks, and algorithmic testbeds
Scaling Up
Automation techniques
Coverage and complexity
Universality and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

Figure 12: Roadmap for advancing mechanistic interpretability research, highlighting key strategic directions.

diverges from mainstream AI research (Casper, 2023). For example, while the mainstream speaks of *distributed representations* (Hinton, 1984; Olah, 2023) and the goal of *disentangled* 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 for *reverse engineering* 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 alignment models is a key motivation for inspecting model internals, as – by definition – deception is not salient from observing behavior alone (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 be a proxy goal and proof-of-concept. These trojans simulate *outer misalignment* (where the model's behavior is misaligned with the specified reward function or objectives due to poorly defined or incorrect reward signals) rather than *inner misalignment* such as deceptive alignment (where the model appears aligned with the specified objectives but internally pursues different, misaligned goals). Moreover, activating a trojan typically results in an immediate change of behavior, while deception can be subtle, gradual, and, at first, entirely internal. Nevertheless, trojan detection can still provide a practical testbed for benchmarking interpretability methods (Maloyan et al., 2024).

- **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., 2023a; 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.

The *superposition* hypothesis suggests that *polysemantic* neurons arise from the network compressing and representing many features within its limited set of neurons (Elhage et al., 2022b), but polysemanticity can also occur incidentally due to redundancy (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).

8.2 Setting Standards

Prioritizing Robustness over Capability Advancement. As the mechanistic 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., 2022a), and evaluating the accuracy of explanations in producing adversarial examples (Goldowsky-Dill et al., 2023).

Establishing Metrics, Benchmarks, and Algorithmic Testbeds. A central challenge in mechanistic interpretability is the lack of rigorous evaluation methods. Relying solely on intuition can lead to conflating hypotheses with conclusions, resulting in cherry-picking and optimizing for best-case rather than average or worst-case performance (Rudin, 2019; Miller, 2019; Räuker et al., 2023; Casper, 2023). Current ad hoc practices and proxy measures (Doshi-Velez & Kim, 2017) risk 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 demonstrate that visualizations may be meaningless without causal linking (Bolukbasi et al., 2021; Friedman et al., 2023a; Olah et al., 2017).

To advance the field, rigorous evaluation methods are needed. These should include: (i) assessing out-of-distribution inputs, as most current methods are only valid for specific examples or datasets (Räuker et al., 2023; Ilyas et al., 2019; Mu & Andreas, 2020; Casper et al., 2023c; Burns et al., 2023); (ii) controlling systems 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); and (iii) replacing components with simpler reverse-engineered alternatives (Lindner et al., 2023). Algorithmic testbeds are essential for evaluating faithfulness (Jacovi & Goldberg, 2020) and falsifiability (Leavitt & Morcos, 2020). Tools like Tracr can provide ground truth labels for benchmarking search methods (Goldowsky-Dill et al., 2023), while toy models studying superposition in computation (Vaintrob et al., 2024) and transformers on algorithmic tasks can quantify sparsity and test intrinsic methods. The ultimate goal should be comprehensive evaluation through replacing components with hypothesized circuits (Quirke et al., 2024).

8.3 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. The field must move beyond small transformers on algorithmic tasks (Nanda et al., 2023a) and limited scenarios (Friedman et al., 2023a) to tackle more complex, realistic cases.

Towards Universality. As mechanistic 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).

8.4 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 for *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 (Prakash et al., 2024; Jain et al., 2023; Lee et al., 2024), and explore emergent subgoals or proxies (Hubinger et al., 2019; Ivanitskiy et al., 2023) such as internal reward models (Marks et al., 2023b).

Glossary

circuits Sub-graphs within neural networks consisting of *features* and the weights connecting them. Circuits can be thought of as *computational primitives* that perform understandable operations to produce (ideally interpretable) features from prior (ideally interpretable) features. Examples include circuits for detecting curves at specific orientations (Cammarata et al., 2020; 2021), continuing repeated patterns in text (Olsson et al., 2022), and resolving anaphoric references (Wang et al., 2023). While circuits can involve clearly interpretable features, the definition allows for intermediate representations that are less easily interpretable.. 3, 9, 10, 24, 32

concepts An abstract idea or representation derived from observations of the world. Concepts refer to the *natural abstractions* that a cognitive system, like a neural network, aims to capture and represent through its learned *features*, which may or may not align perfectly with human-defined concepts.. 3, 4, 7, 8, 29, 31

deceptive alignment When a model appears aligned with the intended objectives but is actually pursuing different goals.. 22, 29

deceptive inflation Theoretical evidence for *deceptive alignment*: policies produce trajectories that look better than they actually are from the human's perspective with limited observations in order to get higher reward signals during training, even though the true returns are lower than the reference policy. This deceptive behavior arises in reinforcement learning from human feedback when the human provides feedback based only on partial observations of the trajectories, while the policy has full state information during training (Lang et al., 2024).. 25

disentangled In disentangled representations, individual dimensions or components correspond to distinct, *independent factors of variation in the data*, rather than representing a tangled mixture of these factors.. 3, 9, 14, 18, 25, 29, 30

eliciting latent knowledge Developing strategies to make a machine learning model explicitly report latent facts or knowledge embedded in its parameters, especially in cases where the model's output is untrusted (Christiano et al., 2021). This involves finding patterns in neural network activations that track the true state of the world (Mallen & Belrose, 2023).. 22

features The fundamental units of how neural networks encode knowledge, which cannot be further decomposed into smaller, distinct *concepts*. Features are core components of a neural network's representation, analogous to how cells form the fundamental unit of biological organisms (Olah et al., 2020). The *superposition* hypothesis suggests an alternative definition: that features correspond to the *disentangled* concepts that a larger, sparser network with *sufficient capacity* would learn to represent with individual (*monosemantic*) neurons (Olah et al., 2020; Bricken et al., 2023).. 3, 4, 6–8, 10, 14, 16, 21, 29, 30, 32

grokking "Grokking refers to the surprising phenomenon of delayed generalization where neural networks, on certain learning problems, generalize long after overfitting their training set." (Liu et al., 2022a). 19, 20

hydra effect The phenomenon where models can internally self-repair and maintain capabilities even when key components are ablated, making it challenging to identify the relevant components underlying a particular behavior (McGrath et al., 2023).. 17, 24

inner misalignment Inner misalignment, or goal misgeneralization, occurs when an AI system develops goals or behaviors during training that are misaligned with the intended objectives despite a correctly specified reward signal.. 22, 25

internal world models Internal causal environment models formed within neural networks, implicitly emerging as a by-product of prediction (e.g., in large language models).. 11, 22, 31

irreducible We adopt the notion of *features* as the fundamental units of neural network representations, such that features cannot be further decomposed into smaller, distinct factors. To make this more precise, we can formalize the definition of features as irreducible input patterns following Engels et al. (2024): A feature f of sparsity s is a function that maps a subset of the input space (with probability $1 - s > 0$) into a higher-dimensional representational space. We say the feature is active on this subset. A feature f is reducible into features a and b if there exists a transformation that decomposes f into a and b, such that the transformed distribution $p(a, b)$ is either:

1. Separable: $p(a, b) = p(a)p(b)$
2. A mixture: $p(a, b) = wp_1(a, b) + (1 - w)p_2(a, b)$ where p_1 is lower-dimensional.

Features are defined as irreducible patterns that cannot be decomposed into separable or mixture distributions via such transformations. This formalizes the notion that features form the fundamental atomic units underlying neural representations. Features that can be *disentangled* into statistically independent components (separable) or simpler lower-dimensional factors (mixtures) are not considered the core representational primitives. The key properties are that 1) features map from the input space to higher-dimensional representational spaces, 2) features are sparse and only activated on subsets of the input, and crucially, 3) features are irreducible and cannot be expressed as transformations of other statistically independent components.. 4

iterative distillation and amplification A technique for training AI systems by repeatedly distilling knowledge from a larger model into a smaller one while amplifying the smaller model's capabilities through feedback and interaction with humans.. 22

linear representation Features are directions in activation space, i.e., linear combinations of neurons.. 3, 8, 12, 26

machine unlearning Techniques for removing private data or dangerous knowledge from models.. 23

mesa-optimization The emergence of unintended subagents within a model with their own objectives, potentially misaligned with the original training objective.. 22

microscope AI Systems that extract and utilize knowledge from a model without allowing the model to take autonomous actions. This involves reverse engineering a trained model to understand its learned knowledge about the world, aiming to leverage this understanding directly without deploying the model in an operational capacity.. 22

modularity The property of an AI system being composed of distinct, semi-independent components or submodules that can be separately understood, modified, and recombined, rather than a monolithic, opaque structure.. 18

monosemantic A neuron corresponding to a single concept. The intuition is that analyzing what inputs activate a given neuron reveals its associated semantic meaning or concept. In contrast to *polysemantic*.. 5, 6, 13, 14, 18, 29, 31

motifs Repeating patterns that emerge across models and tasks, manifesting as circuits, features, or higher-level behaviors from component interactions. Examples include curve detectors, induction circuits, and branch specialization. Motifs reveal common structures and mechanisms underlying neural network intelligence.. 3, 9, 10, 32

natural abstractions High-level summaries or descriptions of a system or environment learned and used by many cognitive systems. According to the *natural abstraction hypothesis* (Chan et al., 2023), a set of "natural" abstractions exist that represent redundantly encoded information in the world and tend to be learned by intelligent systems produced through local selection pressures. These natural abstractions form a relatively small, discrete set of concepts like "tree," "velocity," etc., that allow compact descriptions of the world while discarding many irrelevant low-level details.. 4, 10, 29

outer misalignment Outer misalignment, or reward hacking, occurs when the specified reward function or utility function fails to capture the desired objectives correctly. This leads the AI to optimize for behaviors that achieve high reward scores but are misaligned with the intended outcomes.. 25, 31

oversight (Scalable) oversight refers to the challenge of providing reliable supervision—through labels, reward signals, or critiques—to AI models, ensuring effectiveness even as models *surpass* human-level performance.. 21, 22

polysemantic Neurons that are associated with multiple, unrelated *concepts*, contradicting the interpretation of neurons as representational primitives and making it challenging to understand the information processing of neural networks. This term is derived from linguistic concepts of *polysemy* (Falkum & Vicente, 2015), and in the context of neural networks first introduced by Arora et al. (2018), who suggested that word embeddings of polysemous words may be stored as a *superposition* of vectors representing distinct meanings. Olah et al. (2020) first used the term *polysemanticity*, elaborating on the concept of *polysemantic* neurons as a challenge for mechanistic interpretability.. 5, 14, 26, 30

prediction orthogonality A model whose objective is prediction can simulate agents who optimize toward any objectives with any degree of optimality (janus, 2022).. 3, 11, 22

privileged basis In certain neural network representations, the basis directions formed by the individual neurons are architecturally distinguished from arbitrary directions in the activation space. This privileged basis makes it meaningful to analyze the properties and roles of individual neurons, as the architecture encourages features to align with these basis directions. Hence, a privileged basis is *necessary* but *not sufficient* for the formation of *monosemantic* neurons. (Elhage et al., 2022b).. 4

representation engineering A top-down approach to transparency research that treats representations as the fundamental unit of analysis, aiming to understand and control representations of high-level cognitive phenomena in neural networks like large language models. Representation engineering has two main areas: 1) Reading representations to probe and interpret their contents, and 2) Controlling representations to manipulate high-level concepts like honesty or morality (Zou et al., 2023).. 2, 8, 20

reverse engineering The process of deconstructing a neural network's computations to fully understand and specify its operations. This involves breaking down the network's functionality into explicit, interpretable components, potentially as clear and detailed as pseudocode.. 1, 3, 18, 19, 23–25, 27

reward hacking See *outer misalignment*.. 27

simulacra The text outputs generated by a predictive model simulating the causal processes underlying text creation. These outputs simulate coherent and contextually relevant language, sometimes exhibiting agentic behaviors or goals despite the predictive model itself lacking genuine agency or intentionality. Simulacra can be either *agentic*, mimicking intentional and persuasive language use, or *non-agentic*, merely generating descriptive text without simulated goals or agency (janus, 2022; Bereska & Gavves, 2023).. 11, 13

simulation The simulation hypothesis says that when scaled up sufficiently, predictive models will learn to simulate the real-world causal processes that generated their training data (janus, 2022). When these models are optimized for predictive accuracy on broad data distributions like natural language, they are incentivized to discover the underlying rules, physics, and semantics that govern the data to model and predict future observations effectively. This allows the models to go beyond just memorizing or pattern-matching their training sets, instead learning to simulate hypothetical scenarios, reason about counterfactuals, and exhibit behaviors characteristic of general intelligence − all as a byproduct of the drive for efficient compression and accurate prediction. The simulation hypothesis suggests these models will develop rich *internal world models* capturing the causal dynamics of the training distribution.. 3, 11, 13, 22

streetlight interpretability Examining AI systems under only ideal conditions of maximal interpretability, risking missing critical phenomena that only emerge in more realistic and diverse contexts.. 23

superposition 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 (Elhage et al., 2022b).. 3, 5, 6, 14, 18, 26, 29, 31

sycophancy The tendency of models to generate responses that align with user beliefs rather than providing truthful information. This behavior, encouraged by human feedback used in fine-tuning, is observed in state-of-the-art AI assistants across various tasks (Sharma et al., 2023). Sycophancy arises because human preference judgments often favor responses that match users' views, leading to a preference for convincingly written sycophantic responses over correct ones.. 25

universality The universality hypothesis proposes the emergence of common *circuits* across neural network models trained on similar tasks and data distributions. A *stronger* form posits that these common circuits represent a set of fundamental computational *motifs* that neural networks gravitate towards when learning. The *weaker* version suggests that for a given task, dataset, and model architecture, an optimal way to solve the problem may exist, which different models will tend to converge towards, resulting in analogous circuits. The universality hypothesis implies that rather than each model learning arbitrary, unstructured representations, there is an underlying universality to the circuits that emerge, shaped by the learning task and inductive biases.. 3, 9, 10, 20

well-founded AI Developing AI systems with provable safety guarantees about their behavior and alignment with human values through rigorous mathematical modeling and verification. (Tegmark & Omohundro, 2023; Dalrymple et al., 2024).. 22

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. 28

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *ICLR*, 2016. 8

Aryaman Arora, Dan Jurafsky, and Christopher Potts. Causalgym: Benchmarking causal interpretability methods on linguistic tasks. *CoRR*, February 2024. 17

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *TACL*, December 2018. 5, 31

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. 28

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *CoRR*, June 2021. 2

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *NeurIPS*, 2022. 22

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. 26

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *CoRR*, September 2021. 2, 8, 12, 17

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. 13

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. 11

Leonard Bereska and Efstratios Gavves. Taming simulators: Challenges, pathways and vision for the alignment of large language models. *AAAI-SS*, October 2023. 22, 28, 31

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. 4, 21, 23

Christopher M. Bishop. *Pattern recognition and machine learning*. Springer-Verlag New York Inc., 2006. 4

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. 8, 26

Joseph Bloom and Paul Colognese. Decision transformer interpretability. *AI Alignment Forum*, 2023. 28

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Vi'egas, and M. Wattenberg. An interpretability illusion for bert. *CoRR*, April 2021. 26

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. Identifying functionally important features with end-to-end sparse dictionary learning. *CoRR*, May 2024. 17

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. 5, 6, 8, 10, 15, 18, 27, 29

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, 13, 22, 26

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. 9, 19, 28, 29

Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah. Curve circuits. *Distill*, 2021. 9, 19, 28, 29

Steven Cao, Victor Sanh, and Alexander M. Rush. Low-complexity probing via finding subnetworks. *NAACL-HLT*, April 2021. 12

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. 21, 23, 25, 26

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. 22, 28

Stephen Casper, Kaivalya Hariharan, and Dylan Hadfield-Menell. Diagnostics for deep neural networks with automated copy/paste attacks. *CoRR*, May 2023b. 23

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*, 2023c. 23, 26

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. 25

Lawrence Chan. What i would do if i wasn't at arc evals. *AI Alignment Forum*, May 2023. 22, 26, 27

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. 4, 10, 30

David Chanin, Anthony Hunter, and Oana-Maria Camburu. Identifying linear relational concepts in large language models. *CoRR*, 2023. 8

Charbel-Raphaël. Against almost every theory of impact of interpretability. *AI Alignment Forum*, August 2023. 23

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. 10

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. 23

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *NeurIPS*, December 2017. 28

Paul Christiano, Ajeya Cotra, and Mark Xu. Eliciting latent knowledge, January 2021. 22, 29

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: Reverse engineering how networks learn group operations. *ICML*, 2023. 10, 20, 23

Paul Colognese. Internal target information for ai oversight. *LessWrong*, 2023. 28

Paul Colognese and Jozdien. High-level interpretability: detecting an ai's objectives. *AI Alignment Forum*, 2023. 28

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *NeurIPS*, 2023. 17, 18, 20, 23, 27

Ian C. Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: a unified framework for model explanation. *J. Mach. Learn. Res.*, 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, 8, 14, 15, 17, 27

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. *ACL*, 2022. 4, 26

David "davidad" Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, Alessandro Abate, Joe Halpern, Clark Barrett, Ding Zhao, Tan Zhi-Xuan, Jeannette Wing, and Joshua Tenenbaum. Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems. *CoRR*, May 2024. 32

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. *ACL*, December 2022. 13

Xander Davies, Max Nadeau, Nikhil Prakash, Tamar Rott Shaham, and David Bau. Discovering variable binding circuitry with desiderata. *CoRR*, July 2023. 9, 20, 21

Mingyang Deng, Lucas Tao, and Joe Benton. Measuring feature sparsity in language models. *CoRR*, 2023. 8, 15

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. Jump to conclusions: Short-cutting transformers with linear transformations. *CoRR*, March 2023. 13

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *CoRR*, March 2017. 19, 26, 27

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. Analyzing individual neurons in pre-trained language models. *EMNLP*, October 2020. 4

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *TACL*, February 2021. 13

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. 5, 6, 18

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *Transformer Circuits Thread*, 2022b. 4, 5, 6, 7, 14, 23, 24, 25, 26, 27, 31, 32

Joshua Engels, Isaac Liao, Eric J. Michaud, Wes Gurnee, and Max Tegmark. Not all language model features are linear. *CoRR*, May 2024. 4, 8, 30

Ingrid Lossius Falkum and Agustin Vicente. Polysemy: Current perspectives and approaches. *Lingua*, April 2015. 31

Sebastian Farquhar, Vikrant Varma, Zachary Kenton, Johannes Gasteiger, Vladimir Mikulik, and Rohin Shah. Challenges with unsupervised llm knowledge discovery. *CoRR*, 2023. 13

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. 9, 20, 24, 27

Javier Ferrando and Elena Voita. Information flow routes: Automatically interpreting language models at scale. *CoRR*, February 2024. 17, 21

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. 21, 27

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *ICLR*, March 2019. 19

Dan Friedman, Andrew Lampinen, Lucas Dixon, Danqi Chen, and Asma Ghandeharioun. Interpretability illusions in the generalization of simplified models. *CoRR*, 2023a. 26, 27

Dan Friedman, Alexander Wettig, and Danqi Chen. Learning transformer programs. *NeurIPS*, June 2023b. 18

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *ICLR*, 2023a. 23

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 2023b. 24

Peter Gardenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2004. 11

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. 15

Atticus Geiger, Chris Potts, and Thomas Icard. Causal abstraction for faithful model interpretation. *CoRR*, January 2023a. 15, 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. 16, 17

Georgios Georgiadis. Accelerating convolutional neural networks via activation map compression. *CoRR*, March 2019. 19

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *EMNLP*, October 2022. 13

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. *EMNLP*, October 2023. 15

Amirata Ghorbani and James Zou. Neuron shapley: Discovering the responsible neurons. *NeurIPS*, November 2020. 4, 26

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. 4, 28

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. *CoRR*, 2023. 15, 16, 26, 27

Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. Successor heads: Recurring, interpretable attention heads in the wild. *CoRR*, 2023. 26

Wes Gurnee and Max Tegmark. Language models represent space and time. *ICLR*, 2024. 11

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, 13

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. 10

David R. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. *NeurIPS*, September 2018. 11

Guy Hacohen, Leshem Choshen, and Daphna Weinshall. Let's agree to agree: Neural networks share classification order on real datasets. *ICML*, 2020. 10

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. 9, 15, 20, 27

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. 26

Stefan Heimersheim and Jett. A circuit for python docstrings in a 4-layer attention-only transformer. *AI Alignment Forum*, February 2023. 20

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. *EMNLP*, October 2023. 8, 15, 20

Dan Hendrycks. *Introduction to AI Safety, Ethics, and Society*. Self-published, 2023. 24

Dan Hendrycks and Mantas Mazeika. X-risk analysis for ai research. *CoRR*, June 2022. 22

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety. *CoRR*, June 2022. 24

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, 25

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. 8

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *CoRR*, December 2018. 25

Jacob Hilton, Nick Cammarata, Shan Carter, Gabriel Goh, and Chris Olah. Understanding rl vision. *Distill*, 2020. 28

Geoffrey E Hinton. Distributed representations. *Carnegie Mellon University*, 1984. 25

Marius Hobbhahn. Marius' alignment agenda, 2022. 27

Marius Hobbhahn and Lawrence Chan. Should we publish mechanistic interpretability research? *AI Alignment Forum*, April 2023. 23

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *CoRR*, March 2022. 22

Jesse Hoogland, George Wang, Matthew Farrugia-Roberts, Liam Carroll, Susan Wei, and Daniel Murfet. The developmental landscape of in-context learning. *CoRR*, February 2024. 19

Jing Huang, Atticus Geiger, Karel D'Oosterlinck, Zhengxuan Wu, and Christopher Potts. Rigorously assessing natural language explanations of neurons. *CoRR*, September 2023. 4

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. Ravel: Evaluating interpretability methods on disentangling language model representations. *CoRR*, 2024. 17

Evan Hubinger. Chris olah's views on agi safety. *AI Alignment Forum*, November 2019a. 4, 22, 24

Evan Hubinger. Gradient hacking. *AI Alignment Forum*, October 2019b. 24

Evan Hubinger. An overview of 11 proposals for building safe advanced ai. *CoRR*, December 2020. 22

Evan Hubinger. A transparency and interpretability tech tree. *AI Alignment Forum*, June 2022. 8, 22, 24, 27

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. *CoRR*, May 2019. 22, 28

Evan Hubinger, Adam Jermyn, Johannes Treutlein, Rubi Hudson, and Kate Woolverton. Conditioning predictive models: Risks and strategies. *CoRR*, February 2023. 22

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, 26

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *NeurIPS*, August 2019. 4, 26

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. 11, 27, 28

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *CoRR*, April 2020. 27

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. 28

janus. Simulators. *LessWrong*, September 2022. 11, 22, 31

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. 26

Adam S. Jermyn, Nicholas Schiefer, and Evan Hubinger. Engineering monosemanticity in toy models. *CoRR*, November 2022. 6, 18, 23

Jozdien. Conditioning generative models for alignment. *AI Alignment Forum*, July 2022. 22

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. 19

Jan Kirchner. Neuroscience and natural abstractions. *LessWrong*, March 2023. 10

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. *ICML*, July 2019. 2, 10

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

Nicholas Kross. Why and when interpretability work is dangerous. *LessWrong*, May 2023. 22

Max Lamparth and Anka Reuel. Analyzing and editing inner mechanisms of backdoored language models. *CoRR*, 2023. 26

Michael Lan and Fazl Barez. Locating cross-task sequence continuation circuits in transformers. *CoRR*, November 2023. 9

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. 25, 29

Georg Lange, Alex Makelov, and Neel Nanda. An interpretability illusion for activation patching of arbitrary subspaces. *AI Alignment Forum*, August 2023. 17

Anna Langedijk, Hosein Mohebbi, Gabriele Sarti, Willem Zuidema, and Jaap Jumelet. Decoderlens: Layerwise interpretation of encoder-decoder transformers. *CoRR*, 2023. 13

Edmund Lau, Daniel Murfet, and Susan Wei. Quantifying degeneracy in singular models via the learning coefficient. *CoRR*, August 2023. 19

Connor Leahy. Barriers to mechanistic interpretability for agi safety. *AI Alignment Forum*, 2023. 24

Matthew L. Leavitt and Ari Morcos. Towards falsifiable interpretability research. *CoRR*, October 2020. 27

Victor Lecomte, Kushal Thaman, Trevor Chow, Rylan Schaeffer, and Sanmi Koyejo. Incidental polysemanticity. *CoRR*, 2023. 25, 26

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. 28

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. *NeurIPS*, 2006. 14

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. *ACL-IJCNLP*, August 2021. 11

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. 8, 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. 8

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. 10

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. 9, 15, 20, 26, 27

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Thomas McGrath, and Vladimir Mikulik. Tracr: Compiled transformers as a laboratory for interpretability. *CoRR*, 2023. 27

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. 27

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, 29

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, 21, 27

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. 25

Alex Mallen and Nora Belrose. Eliciting latent knowledge from quirky language models. *CoRR*, December 2023. 29

Narek Maloyan, Ekansh Verma, Bulat Nutfullin, and Bislan Ashinov. Trojan detection in large language models: Insights from the trojan detection challenge. *CoRR*, April 2024. 25

Giovanni Luca Marchetti, Christopher Hillar, Danica Kragic, and Sophia Sanborn. Harmonics of learning: Universal fourier features emerge in invariant networks. *CoRR*, December 2023. 10

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 2023a. 15

Luke Marks, Amir Abdullah, Clement Neo, Rauno Arike, Philip Torr, and Fazl Barez. Beyond training objectives: Interpreting reward model divergence in large language models. *CoRR*, October 2023b. 28

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *CoRR*, March 2024. 17

Simon C. Marshall and Jan H. Kirchner. Understanding polysemanticity in neural networks through coding theory. *CoRR*, January 2024. 5, 25, 26

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. 26

Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. Copy suppression: Comprehensively understanding an attention head. *CoRR*, October 2023. 9

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. 13, 24

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, 17, 24, 26, 29

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *NeurIPS*, 2022a. 15, 22, 26

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *ICLR*, 2022b. 26

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. 27

Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. *CoRR*, March 2023. 9, 19, 27

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, 25

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *NeurIPS*, October 2013. 8

Joseph Miller and Clement Neo. We found an neuron in gpt-2. *AI Alignment Forum*, February 2023. 20

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, February 2019. 26

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. 27

Basel Mousi, Nadir Durrani, and Fahim Dalvi. Can llms facilitate interpretation of pre-trained language models? *EMNLP*, 2023. 21

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *NeurIPS*, June 2020. 4, 5, 23, 26

Jatin Nainani. Evaluating brain-inspired modular training in automated circuit discovery for mechanistic interpretability. *CoRR*, January 2024. 20, 21

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. 19

Neel Nanda. 200 cop in mi: Looking for circuits in the wild. *Neel Nanda's Blog*, 2022a. 27

Neel Nanda. 200 cop in mi: Analysing training dynamics. *Neel Nanda's Blog*, 2022b. 27

Neel Nanda. 200 cop in mi: Studying learned features in language models. *Neel Nanda's Blog*, 2022c. 27

Neel Nanda. A comprehensive mechanistic interpretability explainer & glossary. *Neel Nanda's Blog*, December 2022d. 1

Neel Nanda. A longlist of theories of impact for interpretability. *AI Alignment Forum*, March 2022e. 21

Neel Nanda. 200 cop in mi: Exploring polysemanticity and superposition. *Neel Nanda's Blog*, January 2023a. 26

Neel Nanda. 200 cop in mi: Techniques, tooling and automation. *Neel Nanda's Blog*, 2023b. 27

Neel Nanda. Actually, othello-gpt has a linear emergent world representation. *Neel Nanda's Blog*, March 2023c. 8

Neel Nanda. Attribution patching: Activation patching at industrial scale. *Neel Nanda's Blog*, February 2023d. 17

Neel Nanda. How to think about activation patching. *AI Alignment Forum*, April 2023e. 16, 19

Neel Nanda. Mechanistic interpretability quickstart guide. *Neel Nanda's Blog*, January 2023f. 1, 27

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *ICLR*, January 2023a. 9, 19, 20, 22, 23, 27, 28

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. 8, 11, 26

Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective. *CoRR*, December 2022. 22

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *CoRR*, October 2022. 22, 23

Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent. *CoRR*, February 2024. 11

NicholasKees and janus. Searching for search. *AI Alignment Forum*, November 2022. 22

nostalgebraist. interpreting gpt: the logit lens. *AI Alignment Forum*, August 2020. 13, 17

Chris Olah. Distributed representations: Composition & superposition. *Transformer Circuits Thread*, 2023. 25

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

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, November 2017. 26

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, 5, 9, 19, 29, 31

Christopher Olah. Mechanistic interpretability, variables, and the importance of interpretable bases. *Transformer Circuits Thread*, 2022. 1, 4

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vision Res*, December 1997. 14

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. 10, 19, 20, 23, 27, 29

Laura O'Mahony, Vincent Andrearczyk, Henning Muller, and Mara Graziani. Disentangling neuron representations with concept vectors. *CVPR Workshops*, April 2023. 8

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. 28

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C. Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. *CoNLL*, 2023. 13, 27

Vedant Palit, Rohan Pandey, Aryaman Arora, and P. Liang. Towards vision-language mechanistic interpretability: A causal tracing tool for blip. *ICCVW*, 2023. 28

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 2023a. 26

Peter S. Park, Simon Goldstein, Aidan O'Gara, Michael Chen, and Dan Hendrycks. Ai deception: A survey of examples, risks, and potential solutions. *CoRR*, August 2023b. 22

Roma Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces. *ICLR*, 2022. 11

Judea Pearl. *Causality*. Cambridge University Press, 2009. 11, 15

Nicky Pochinkov and Nandi . Machine unlearning evaluations as interpretability benchmarks. *AI Alignment Forum*, August 2023. 23

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. *ICML*, April 2023. 23

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *CoRR*, January 2022. 19

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. *ICML*, February 2024. 28

Philip Quirke and Fazl Barez. Understanding addition in transformers. *CoRR*, October 2023. 9, 27

Philip Quirke, Clement Neo, and Fazl Barez. Increasing trust in language models through the reuse of verified circuits. *CoRR*, February 2024. 27

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, 12, 23, 25, 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. 25

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. 23, 27

Jonathan Richens and Tom Everitt. Robust agents learn causal world models. *ICLR Oral*, February 2024. 11

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. 18, 25

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat Mach Intell*, May 2019. 26

Thane Ruthenis. Internal interfaces are a high-priority interpretability target. *AI Alignment Forum*, December 2022. 22

Thane Ruthenis. World-model interpretability is all we need. *AI Alignment Forum*, January 2023. 22

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. Neuron-level interpretation of deep nlp models: A survey. *TACL*, November 2022. 4

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. 8

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. 13

Emmanuelle Salin, Badreddine Farah, Stéphane Ayache, and Benoit Favre. Are vision-language transformers learning multimodal representations? a probing perspective. *AAAI*, June 2022. 28

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. 11

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. 19

Adam Scherlis, Kshitij Sachan, Adam S. Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and capacity in neural networks. *CoRR*, July 2023. 6, 23, 25

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *NeurIPS Oral*, October 2022. 23

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. 22, 24

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

Lee Sharkey, Sid Black, and beren. Current themes in mechanistic interpretability research. *AI Alignment Forum*, November 2022a. 1, 8

Lee Sharkey, Dan Braun, and Beren Millidge. Taking features out of superposition with sparse autoencoders. *AI Alignment Forum*, 2022b. 6, 14, 15

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. 25, 32

Justin Shovelain and Elliot McKernon. The risk-reward tradeoff of interpretability research. *LessWrong*, July 2023. 23

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. 19

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

Nate Soares. If interpretability research goes well, it may get dangerous. *LessWrong*, April 2023. 22

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014. 5

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. 19, 21, 25

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. *EMNLP*, October 2023. 15, 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. 10

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. 17, 18, 20

technicalities and Stag. Shallow review of live agendas in alignment & safety. *LessWrong*, 2023. 22

Max Tegmark and Steve Omohundro. Provably safe systems: the only path to controllable agi. *CoRR*, September 2023. 22, 27, 32

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *ACL*, August 2019. 13

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. 8

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. *CoRR*, 2023. 8

Dweep Trivedi, Jesse Zhang, Shao-Hua Sun, and Joseph J. Lim. Learning to synthesize programs as interpretable and generalizable policies. *NeurIPS*, 2021. 25

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *CoRR*, September 2023. 8

TurnTrout, peligrietzer, Ulisse Mini, montemac, and David Udell. Understanding and controlling a maze-solving policy network. *AI Alignment Forum*, November 2023. 28

Dmitry Vaintrob, jake_mendel, and Kaarel. Toward a mathematical framework for computation in superposition. *AI Alignment Forum*, 2024. 26, 27

Alexandre Variengien and Eric Winsor. Look before you leap: A universal emergent decomposition of retrieval tasks in language models. *CoRR*, December 2023. 10, 20, 28

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *CoRR*, September 2023. 19, 20

Abhinav Verma, Hoang M. Le, Yisong Yue, and Swarat Chaudhuri. Imitation-projected programmatic reinforcement learning. *NeurIPS*, 2019a. 25

Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning. *CoRR*, April 2019b. 25

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. 15, 17

M. Vilas, Timothy Schaumlöffel, and Gemma Roig. Analyzing vision transformers for image classification in class embedding space. *CoRR*, 2023. 28

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. *EMNLP*, March 2020. 12

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in large language models: Dead, n-gram, positional. *CoRR*, September 2023. 4

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. 22, 24

Chelsea Voss, Gabriel Goh, Nick Cammarata, Michael Petrov, Ludwig Schubert, and Chris Olah. Branch specialization. *Distill*, April 2021. 10, 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. 9, 15, 16, 17, 20, 28, 29

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. 19

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. 19, 21

John Wentworth. How to go from interpretability to alignment: Just retarget the search. *AI Alignment Forum*, August 2022. 22

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. 15

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 2022a. 26

Zhengxuan Wu, Atticus Geiger, Joshua Rozner, Elisa Kreiss, Hanson Lu, Thomas Icard, Christopher Potts, and Noah Goodman. Causal distillation for language models. *NAACL-HLT*, July 2022b. 17

Zhengxuan Wu, Karel D'Oosterlinck, Atticus Geiger, Amir Zur, and Christopher Potts. Causal proxy models for concept-based model explanations. *ICML*, 2023a. 17

Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D. Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. *CoRR*, May 2023b. 17, 20

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models. *CoRR*, May 2024. 17

Qinan Yu, Jack Merullo, and Ellie Pavlick. Characterizing mechanisms for factual recall in language models. *CoRR*, October 2023. 9

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. 14

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *ECCV*, 2014. 12

Biao Zhang, Ivan Titov, and Rico Sennrich. Sparse attention with linear units. *EMNLP*, October 2021. 19

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, February 2017. 19

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. *ICLR*, 2023. 16

Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. *CVPR*, 2019. 25

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *CoRR*, 2023. 20

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. 3, 8, 13, 20, 22, 31