Interventionist Methods for Interpreting Deep Neural Networks

For *Neurocognitive Foundations of Mind*

Raphaël Millière
Department of Philosophy
Macquarie University
raphael.milliere@mq.edu.eu

Cameron Buckner
Philosophy Department
University of Houston
cjbuckner@uh.edu

Abstract

Recent breakthroughs in artificial intelligence have primarily resulted from training deep neural networks (DNNs) with vast numbers of adjustable parameters on enormous datasets. Due to their complex internal structure, DNNs are frequently characterized as inscrutable "black boxes," making it challenging to interpret the mechanisms underlying their impressive performance. This opacity creates difficulties for explanation, safety assurance, trustworthiness, and comparisons to human cognition, leading to divergent perspectives on these systems. This chapter examines recent developments in interpretability methods for DNNs, with a focus on interventionist approaches inspired by causal explanation in philosophy of science. We argue that these methods offer a promising avenue for understanding how DNNs process information compared to merely behavioral benchmarking and correlational probing. We review key interventionist methods and illustrate their application through practical case studies. These methods allow researchers to identify and manipulate specific computational components within DNNs, providing insights into their causal structure and internal representations. We situate these approaches within the broader framework of causal abstraction, which aims to align low-level neural computations with high-level interpretable models. While acknowledging current limitations, we contend that interventionist methods offer a path towards more rigorous and theoretically grounded interpretability research, potentially informing both AI development and computational cognitive neuroscience.

1. Introduction

The maturation of connectionist models into modern deep neural networks has sparked renewed interest in the computational foundations of cognition (Buckner 2019). This development is particularly evident in the progress of large language models (LLMs) like GPT-4 (OpenAI 2023). While the simpler artificial neural networks that preceded the advent of deep learning were initially inspired by abstract features of brain structure, recent architectural innovations in DNNs are not biologically motivated. Nonetheless, DNNs' impressive performance on complex tasks, when rigorously evaluated under controlled conditions, can inform comparisons with human cognitive abilities (Millière 2024). This parallel has led to a growing trend in cognitive neuroscience of using DNNs as tools for probing

neural computation and representation (Doerig et al. 2023). However, the utility of such comparisons arguably hinges on our ability to interpret the internal workings of DNNs, to address ongoing concerns about their status as neurocognitive models (Räz & Beisbart 2022). Just as the brain remains a "black box" in many respects, information processing within DNNs is typically opaque, despite our having full access to their architecture and weights. This opacity presents both a challenge and an opportunity: by developing methods to uncover the computational and representational structures within DNNs, we may not only enhance our understanding of these artificial systems but also refine our approaches to investigating biological neural networks. In this respect, the development of rigorous interpretability techniques for artificial neural networks may inform future methodological approaches in computational cognitive neuroscience. In this chapter, we focus on interventionist methods for interpreting DNNs, with a particular emphasis on LLMs. First, we review concerns about many common methods used to evaluate and understand LLM performance, such as benchmarking and probing (section 2). Second, we argue that more recent methods based upon increasingly sophisticated and hypothesis-driven interventions on the internal workings of LLMs can address some of these concerns, allowing us to gain a more robust understanding of the internal organization of LLMs (section 3).

2. Challenges to previous interpretation methods

LLMs have made rapid progress on producing human-like linguistic behavior in many different scenarios that were challenging for previous methods in artificial intelligence. They can readily produce grammatically-correct and generally semantically-coherent text. They can respond flexibly to a wide range of questions, including various aptitude and proficiency tests that prove challenging even for educated adults (OpenAI 2023). Moreover, LLMs can produce functional blocks of computer code in multiple programming languages, as well as code in visual markup languages that create coherent images based on textual descriptions (Bubeck et al. 2023). More broadly, their performance on a diverse range of behavioral benchmarking tasks designed to assess specific abilities is converging towards matching or exceeding human baselines. However, there remain significant concerns that these achievements may have less profound explanations they first appear to merit.

One straightforward concern about all of these performance achievements is that any apparently intelligent behavior produced by a DNN-based system could be due to mere memorization of training data. We have elsewhere argued that memorization serve as a useful "null hypothesis" for evaluating the capacities of AI systems (Millière & Buckner 2024). For example, many of GPT-4's feats could in principle be produced by an inefficient and inflexible memory retrieval operation. GPT-4's training set likely encompasses trillions of tokens[1] in millions of textual documents, a significant subset of the whole internet. These training sets include dialogues generated by hundreds of millions of individual humans and hundreds of thousands of academic publications covering potential question-answer pairs. Empirical studies have discovered that the many-layered architecture of DNNs grants them an astounding capacity to efficiently encode their training data, which can allow them to retrieve the right answers to millions of randomly-labeled data points in artificially-constructed datasets where we know a priori there are no abstract principles governing the correct answers (Zhang et al. 2021). This suggests that GPT-4's responses could be generated by approximately–and, in some

[1] "Tokens" are the basic lexical units — whole words or parts of words — used to encode linguistic sequences; for example "camembert" might be tokenized as "cam"+ "emb"+ "ert". Details about the training data and model size of proprietary LLMs like GPT-4 are not publicly available, although we can turn to other LLMs for clues. For example, PaLM 2 has 340 billion parameters and was trained on 3.6 trillion tokens (Anil et al. 2023), while the largest version of Llama 2 has 70 billion parameters and was trained on 2 trillion tokens (Touvron et al. 2023). GPT-4 is rumored to have well over a trillion parameters (Karhade 2023).

cases, exactly–reproducing samples from its training data (McCoy et al. 2023).[2] Compare this to a human student who had found a test's answer key on the Internet and reproduced its answers without any deeper understanding; such regurgitation would not be good evidence that the student was intelligent or understood the subject material. For these reasons, "data contamination" – when the training set contains the very question on which the LLM's abilities are assessed – is considered a serious concern in any report of an LLM's performance, and many think it must be ruled out by default when comparing human and LLM performance (Aiyappa et al. 2023).

While we have argued that skeptical interpretations appealing to mere memorization cannot account for the full range of behaviors exhibited by these models (Millière & Buckner 2024), our previous analysis did not provide a positive account of the mechanisms that might enable LLMs to achieve such breakthrough performance. Providing such an account is challenging due to familiar concerns about the opacity of neural networks, which is exacerbated by LLMs' relatively novel architecture, their enormous number of adjustable parameters, and the sheer magnitude of their training data. To address these challenges, the research community has developed new methods of evaluating the performance of these systems and probing their internal organization, which we explore in this section.

2.1. Mechanistic explanation

In science, mechanistic explanations aim to reveal the causal structure underlying a phenomenon of interest by describing the organized entities and activities that are responsible for producing or maintaining that phenomenon (Machamer et al. 2000). More precisely, a mechanistic explanation identifies the component parts of a mechanism, characterized by their properties and capacities, the causal interactions between these component parts, and the organization of the parts and activities such that they give rise to the phenomenon. Such explanations stand in contrast to purely descriptive or phenomenological models that simply re-describe the phenomenon itself, as well as covering-law explanations that explain by subsuming the phenomenon under empirically discovered regularities or governing laws. Mechanisms explain by revealing how the phenomenon arises from the causal structure of the system, beyond merely capturing the empirical regularities in which it partakes. As such, mechanistic explanations reveal opportunities for manipulation and control over the phenomenon in a way that descriptive or law-based explanations do not.

Understanding the behavior of a simple system like a mechanical clock is easy enough – one can simply open it up and observe the mechanism at work. This is not so straightforward with more complex systems, like the weather, the brain, or artificial neural networks. Neural networks are often described as "black boxes" precisely because the causal mechanisms that explain their behavior seem opaque to scrutiny. As we emphasized elsewhere (Millière & Buckner 2024), knowing the learning objective, architecture, or size of LLMs is insufficient to explain their remarkable performance on challenging tasks, and to determine what functional capacities can be meaningfully ascribed to them. In principle, one could even provide a complete mathematical description of an LLM as a giant composite function, consisting in an absurdly complex sequence of linear and nonlinear transformations across many layers; but such a description, on its own, would be useless to provide a genuine explanation of the network's behavior in specific contexts.[3] The "black box" metaphor

[2] This concern is highlighted by lawsuits against OpenAI, notably from the New York Times (Grynbaum & Mac 2023). These cases document instances where LLMs like GPT-4 have been shown to reproduce substantial portions of copyrighted text verbatim, raising questions about the originality of their outputs.

[3] It is very common to refer to "linear" and "nonlinear" operations in machine learning. Briefly, a linear operation consists of an additive combination of a discrete set of features or inputs where each increase of magnitude in a feature increases the composite function in a constant way, and a non-linear operations involve non-additive or variable influences of features. For example, multiple linear regression functions are linear, whereas functions which involve threshholding

underscores this chasm: it highlights the difficulty to trace precise causal pathways in the network through which specific inputs are transformed into specific outputs. This is why merely *re-describing* what an LLM does in terms of next-token prediction or matrix multiplication – which we have elsewhere called the *re-description fallacy* – cannot possibly settle philosophical debates that are fundamentally about causal organization.

The search for causal mechanisms has become central across the life sciences and cognitive science. Like their artificial counterparts, biological neural networks are "black boxes"; yet neuroscientists are engaged in the project of uncovering multilevel mechanisms underlying psychological capacities and nervous system functions (Craver 2007, Piccinini 2020). The notion of intervention is central to this explanatory project. In the philosophy of science, interventionism holds that causal relationships are best understood in terms of what would change under interventions or manipulations to parts of the system (Woodward 2005). More specifically, X is considered a direct cause of Y if and only if there exists a possible intervention on X that will change Y (or the probability distribution of Y) when all other variables in the system are held fixed. This relationship is asymmetric – intervening on Y does not change X if the causal arrow truly points from X to Y. Interventionism eschews regularity or correlational notions of causation, recognizing that a system's behavior depends on more than merely observing regular successions of events. In the context of mechanistic explanation, interventions involve altering specific parts of a posited mechanism piecemeal in order to learn about their causal contribution to the target phenomenon. For example, pharmacology targets specific receptor mechanisms or cell signaling pathways; brain stimulation techniques activate or inhibit activity in restricted brain areas; knockout models remove genes hypothesized to be critical for mechanisms; and optogenetics uses light to activate neurons genetically modified to express light-sensitive channels.

A similar interventionist approach can be applied to unravel causal mechanisms in artificial neural networks like LLMs. Of course, there are major disanalogies between biological nervous systems evolved over millions of years and artificial neural networks designed by human engineers. Nevertheless, the motivation of interventionist research is similar: to achieve explanatory understanding by revealing multilevel mechanisms, not merely observing input-output patterns. Like neuroscientists, computer scientists can aim for explanatory understanding linking particular components of neural networks and patterns of internal activations to specific capacities, such as translating between languages or answering arithmetic questions. This explanatory project is often described as *mechanistic interpretability*. In a broad sense, mechanistic interpretability could be characterized as the search for mechanistic explanations of the behavior of deep neural networks, including LLMs. As we shall see, however, the phrase is often used in a more restrictive sense, to denote a particular set of theoretical assumptions and intervention methods used to achieve mechanistic explanations in machine learning.[4]

2.2. Opening up the black box

There are three main methodological approaches used to investigate the inner structure of neural networks: probing, attribution, and causal intervention. Probing involves training a separate supervised classifier, also known as a diagnostic probe, to predict certain features of the input (e.g., part-of-speech

(the effect of a feature changes above a certain threshold value) or fitting exponential curves to data are non-linear. Linear operations are simpler, more predictable, and require less training data to fit, whereas non-linear operations are more complex, unpredictable, and require more training data, but the latter can also predict more complex patterns in data. Most currently successful machine learning architectures involve complex combinations of linear and non-linear operations.

[4]The lack of a common lexicon in interventionist research on neural networks is rather unfortunate – many different labels exist for similar ideas and methods, with "interpretability" being a particularly confusing term (Lipton 2018). In what follows, we will attempt to give a cohesive overview of this fragmented literature.

tags, dependency relations) from the model's internal activations (Alain & Bengio 2018, Hupkes et al. 2018). High accuracy in decoding a particular linguistic feature F from a probe tuned to a pattern of activation A in a subset of the network can be though to provide evidence that A is sensitive to F, and provide information about the presence of F-information for downstream processing. However, a probe's successful prediction of a certain linguistic feature from a model's activations does not necessarily mean that the feature plays a causal role in the model's behavior (Belinkov 2022). By way of analogy, suppose you throw block letters spelling a word into a pond. With the right apparatus, you might be able to decode the word's identity from the ripples formed by letters over the pond's surface; but this does not provide evidence that the pond represents (let alone "understands") the word in any meaningful sense. It merely shows that the block letters created surface patterns whose origin could be recovered by a sufficiently powerful decoding method. In addition, probes can pick up on spurious correlations, thereby failing to distinguish between genuine representation and incidental associations (Hewitt & Liang 2019).

Methods that do not involve training a separate classifier – also known as nonparametric methods – have been explored as an alternative to probing (Yousefi et al. 2024). For example, one can directly investigate the weights (or attention scores) that attention heads[5] place on different parts of the input sequence. A high attention score on a particular word might be thought to provide evidence that the head is playing a role in processing the semantic or syntactic role of that word in the rest of the sentence. This approach to interpretability falls within the broader and somewhat loose umbrella of "attribution methods" in deep learning. Attribution methods assign importance scores to input features to explain individual model predictions; in other words, they are meant to identify parts of the input that are most influential for the output. While attention patterns in Transformers – the architecture of nearly all recent LLMs like BERT, Claude, Llama, and the GPT-series – intuitively provide clues about the relevance of each input token to model predictions, simply analyzing these patterns in a given layer of the network only offers limited explanatory power (Chefer et al. 2021). In particular, visualizing attention patterns is thoroughly insufficient to draw strong conclusions about the flow of information through the network. This fact is further illustrated by the empirical observation that many attention heads in trained LLMs appear to be redundant or superfluous, in that they can be deactivated without much observable effect on important kinds of performance in the network (Voita et al. 2019, Bian et al. 2021, He et al. 2024, Bansal et al. 2023, Gromov et al. 2024). Ultimately, a common question about probing and attribution methods is whether they provide evidence which is strong enough to disconfirm false hypotheses about the representational organization of these networks. If we want to understand what information the model represents in the input, and how that information is processed through step-by-step computations across layers to drive output predictions, we need to intervene directly on the network to reveal its causal structure.

[5] Self-attention is a key mechanism that distinguishes Transformer-based language models from previous architectures. In essence, self-attention allows each token's representation to be dynamically influenced by other tokens in the input sequence. The process works as follows: (1) for each token, three vectors are computed – a "query", "key", and "value" vector; (2) the attention weight for each pair of tokens is calculated as the dot product of the query vector of one token with the key vector of another, scaled and normalized (via softmax); (3) the output for each token is a weighted sum of all tokens' value vectors, using these attention weights. Multiple attention heads operate in parallel, each potentially focusing on different aspects of the relationships between tokens. These heads are stacked in multiple layers, allowing the model to capture complex hierarchical relationships. The effectiveness of self-attention stems from its ability to model long-range dependencies and its computational efficiency. It has demonstrated success in capturing various syntactic and semantic relationships, contributing significantly to the performance improvements seen in Transformer models. Recent research has uncovered specialized behaviors in attention heads, such as specific syntax-related functions, which together enable the model to perform sophisticated language generation tasks.

3. Interventionist methods for understanding LLMs

To assert that a certain activation pattern in a model genuinely represents a given feature, mainstream philosophical theories of representation require that three criteria be satisfied: (a) the activation pattern should carry information about the feature; (b) the activation pattern should influence the model's behavior in a task-relevant way, and (c) the model should be capable of misrepresenting the feature (Harding 2023). Standard probing methods only provide evidence for the first criterion. Showing that some information about a feature is *actually* used by the LLM to generate its outputs requires suitable *interventions* on patterns of activation encoding that information. Changes in model behavior caused by such interventions should be consistent with the hypothesis that the model represents the target feature, but should not be explainable by mere perturbations from the training set. Accordingly, an increasingly large body of work uses targeted intervention methods to establish causal relationships between language models' internal representations and their behavior.

3.1. From ablation to causal probing

The simplest kind of intervention on neural networks – both artificial and biological – is an ablation. In the context of artificial neural networks, ablation involves disabling or eliminating individual neurons or groups of neurons within a trained model to observe the resulting changes in behavior. By ablating neurons and observing the resulting change in performance metrics such as classification accuracy or reconstruction error, researchers can determine how much each neuron or group of neurons contributes to the network's overall function. Neurons that are critical to network performance will result in a substantial decline in metrics when ablated.

Indiscriminate ablations, common in the early days of connectionist research, involved disabling nodes at random. This approach was aimed at demonstrating general properties of neural networks. One key finding from such studies was the concept of graceful degradation: like biological brains, the performance of neural networks tends to decline in a gradual, rather than abrupt, manner when parts of the network are damaged or disabled (Sejnowski & Rosenberg 1987, Smolensky 1988).

Targeted ablations, on the other hand, involve disabling specific nodes or modules believed to serve distinct representational or functional roles within the network. By analyzing the network's altered behavior, researchers could infer the impact of the disabled nodes, thereby gaining insights into their hypothesized functions (Meyes et al. 2019). This method mirrors a common approach in neuroscience where the study of natural or induced brain lesions helps to unravel the functions of specific brain areas.

However, both indiscriminate and targeted ablation studies are somewhat limited in their ability to fully uncover representational or functional roles in neural networks. This limitation is partly due to the nature of how information is represented within these networks. Traditional views centered around *localized* representations, positing that specific neurons or small groups of neurons could be responsible for representing distinct, complex stimuli or properties. The classic example in neuroscience is the hypothetical 'grandmother cell' – a neuron that would activate exclusively in response to the mental image or property of one's grandmother. While intuitively appealing, this model of localized representation has largely been abandoned in favor of a model on which information about meaningful features is distributed across multiple network components (Plaut & McClelland 2010, Barwich 2019).

In practice, individual neurons in neural networks often encode information about multiple properties or concepts, a phenomenon known as "polysemanticity". This concept has been studied since the early days of connectionist research under the name of distributed representations (Smolensky

1986, Rumelhart et al. 1987). When features are represented as directions in the network's activation space, they may not align with individual neurons (the standard basis of the vector space). In such cases, each neuron captures a mixture of underlying features, resulting in polysemantic neurons. This is characteristic of distributed representations, where features are encoded by the activations of multiple neurons.

More recently, researchers have focused on a specific phenomenon called "superposition", as described by Elhage et al. (2022). Superposition occurs when neural networks represent *more* features than they have neurons or dimensions. In this scenario, the network is forced to encode multiple features in overlapping directions, leading to interference between features. This inherently creates polysemantic neurons, as each neuron must encode information about multiple features to accommodate the excess of features relative to dimensions. Superposition allows neural networks to efficiently encode a large number of sparse features, but it also introduces challenges for interpreting individual neurons and understanding the network's internal representations.

The distributed model of representations in neural networks reflects a more integrated and holistic approach to neural processing, where information is not stored in isolation but as part of a dynamic, interconnected system. The shift from localized to distributed representation models has significant implications for interpreting the results of ablation studies. In a distributed system, disabling a node or a set of nodes can impact the network-wide interplay of neural activities. Therefore, determining the specific effects of localized damage on the overall behavior of the network becomes increasingly complex (Jonas & Kording 2017).

Fortunately, the level of control we have over artificial neural networks allows for far more sophisticated causal interventions that can allow us to "edit" distributed representations in ways consistent with representational or functional hypotheses about their meanings, and then verify corresponding changes in behavior. For example, a more sophisticated approach called "iterative nullspace projection" was developed by Ravfogel et al. (2020). Iterative nullspace projection can determine whether some particular information is causally involved in a neural network's predictions by identifying and removing that information from distributed neural representations, and then assessing the consequence on model behavior. The approach involves iteratively projecting neural representations onto the "nullspace" of a probe – that is, the subspace of the model's activations for which the probe makes the same constant prediction with respect to that feature–to remove detectable information about user-defined target properties.

More specifically, the first step is to train linear probes on internal neural representations to predict values of the property of interest. For example, probes could be trained to classify grammatical number (e.g. whether a noun is plural or singular in Romance languages) from the hidden state of a language model. Projecting representations onto these nullspaces removes information that linearly correlates with property values while preserving unrelated information. After projecting onto the first probe's nullspace, a second probe is trained on the transformed representations to predict the target property. This process is repeated iteratively, with projections onto each new probe's nullspace removing additional detectable information about the target property. Finally, the network makes predictions using the fully projected representations where linear information about the target property has putatively been eliminated. If performance degrades, the target property can be inferred to be causally significant for model performance, suggesting that the network exploits it for its original computations.

Suppose we want to remove grammatical number information from the embedding space of a word embedding model. We would first train a linear probe to predict whether a word is singular or plural from its embedding. This probe could be taken to identify a "grammatical number" direction in the geometry of the LLM's encoding space. We then project all word embeddings onto the nullspace

of this probe, removing the number information. Next we train a second probe to again predict grammatical number on the projected embeddings and project onto its nullspace, and so on iteratively. The key advantage of this iterative approach is that multiple orthogonal directions may encode relevant information about the property of interest. By repeating the process, we are able to identify and strip out multiple directions rather than just the most prominent ones. Once the dimensions are identified, vectors could also be moved in the opposite direction to enhance the information contained about the relevant property, rather than remove it.

For example, Ravfogel et al. (2021) investigated whether information about relative clause boundaries is used by language models like BERT to predict subject-verb agreement across a relative clause. They trained a set of linear probes to predict whether a word is inside a relative clause, based on the model's contextual word embeddings. Each probe defines a direction in the representation space that separates words inside relative clauses from words outside relative clauses. Then, iterative nullspace projection is used to generate 'counterfactual representations' for the masked verb token. The representation is projected into the relative clause feature subspace, and then flipped to the opposite side of the separating hyperplane, either towards the side containing relative words ('positive counterfactual') or away from that side ('negative counterfactual'). This process minimally modifies the representation to incorrectly encode that the word is inside or outside a relative clause. Finally, the effect of the counterfactual representations on the model's number agreement predictions is measured. If swapping in the positive counterfactual increases error rate and the negative counterfactual decreases error rate, this alignment with predictions from linguistic theory suggests the model uses relative clause boundary information appropriately for agreement, confirming the causal effect (fig. 1).

This method is directly inspired by counterfactual approaches to causal explanation in philosophy of science mentioned above (Woodward 2005). Such approaches aim to isolate the causal contribution of some factor X to an outcome Y by minimally altering the value or presence of information about X in the system, while holding all other factors fixed. This allows assessment of whether and how much the factor X is exploited causally by the system to produce Y. The removal of detectable information about a target variable through iterative nullspace projection is analogous to a hypothetical intervention that breaks the links between that variable and the system while leaving other causal mechanisms intact (i.e., the vectors' other meaningful dimensions). The comparison between original model outputs and outputs based on projected representations missing information about X mirrors the evaluation of interventionist counterfactuals, which aims to answer questions such as "What would happen to Y if an intervention prevented information about X from influencing the mechanism?".

3.2. Features and circuits

Intervention methods such as iterative nullspace projection can help us determine whether some target property, such as a particular syntactic feature, is represented by a language model and causally efficacious in its behavior. But this is insufficient to understand language models and other neural networks the way we understand classical computer programs. Indeed, the guiding ideal of interpretability research is to model the internal causal structure of neural networks, such that we can explain some behavior of interest in terms of a series of computational steps applied on the input. Mechanistic interpretability refers to this concerted effort to reverse engineer the internal computations performed by artificial neural networks. Rather than focusing solely on interpreting individual model predictions, it aims to provide a detailed and systematic understanding of how the model transforms inputs to outputs (Elhage et al. 2021). The overarching goal of mechanistic interpretability is to open up the "black box" of neural networks by providing human-intelligible descriptions of the functional modules that drive the emergence of model behaviors.

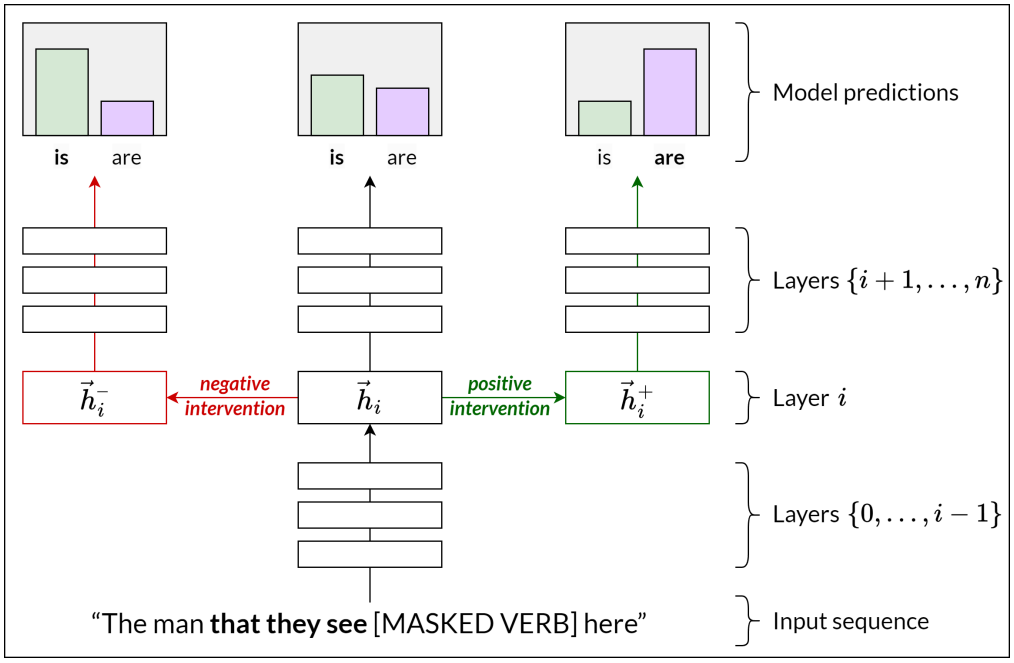

Figure 1 | **Iterative nullspace projection**. Given the representation $\vec{h}_i$ of a masked word in layer i of a Transformer model, a probe is trained to predict relative clause boundaries. The probe's nullspace, which encodes information not relevant to relative clause boundaries, is identified. Two counterfactual representations, $\vec{h}_i^-$ and $\vec{h}_i^+$, are derived by projecting $\vec{h}_i$ onto the nullspace and then performing negative and positive interventions, respectively, along the probe's decision boundary. $\vec{h}_i^-$ encodes that the word is outside a relative clause, while $\vec{h}_i^+$ encodes that it is inside a relative clause, with other information preserved. The model's predictions and using these counterfactual representations are compared to its original prediction to assess the causal effect of the relative clause boundary information on the model's behavior in number agreement.

Like systems neuroscience, mechanistic interpretability specifically targets the algorithmic level of analysis (Marr 1982, Lindsay & Bau 2023). An algorithmic explanation elucidates the flow of information through a network and the sequence of operations performed upon it. Crucially, it abstracts away from engineering (or biological) details, instead focusing on the computations performed by the system. Algorithmic explanations support counterfactual inferences about how changing inputs or network components would affect computations and outputs. This involves identifying structural elements like motifs and circuits that are reusable, such that manipulating them produces systematic effects across many inputs and conditions.

Mechanistic interpretability specifically seeks to reverse-engineer neural networks in terms of learned *features* and reusable *circuits* that operate on those features. Features refer to human-interpretable properties of the input data that the model encodes internally. For computer vision models, such features might be edges, textures, or shapes. For language models, features may correspond to part-of-speech tags, named entities, or semantic relationships. By studying a network's internal activations, researchers aim to determine which features are encoded where.

Circuits, in turn, are chains of operations that detect certain input features and transform them into output features. The goal is to decompose an entire neural network into a hierarchy of understandable features and circuits that process information step-by-step. This modular view would explain how

trained models transform inputs to outputs via learned features and program-like circuits, providing a form of algorithmic understanding.[6]

More formally, a given neural network can be modeled as a graph (sometimes called a computational graph) whose nodes correspond to components of the network at some level of granularity – such as attention heads or individual neurons. The directed edges between nodes represent the flow of information and computations in the neural network. The connectivity defined in the computational graph needs to faithfully represent the actual computation flow in the neural network. In this model, a circuit is a subgraph within the overall computational graph that implements some specific behavior or functionality of interest. The goal of representing the neural network as a computational graph is to systematically analyze it and localize particular circuits that are causally responsible for certain model behaviors.

Once candidate circuits are found, their functionality must be validated through causal interventions. This often involves surgically replacing components of suspected circuits and observing the effect on model outputs. Validated circuits can then be composed into a hierarchical understanding of the succession of transformations enacted by the model. The resulting mechanistic explanation should provide significant behavioral control, for instance allowing targeted editing of model computations.

The picture that emerges from mechanistic interpretability research is that Transformer models can be viewed as containing parallel processing streams (also known as "residual streams"), one at each input token position (fig. 2). While each residual stream encodes information in a very high-dimensional vector space, attention heads at each layer operate over much smaller (low-dimensional) subspaces of the stream that may not overlap with one another. Making use of these disjoint subspaces allows Transformers to route information about tokens and their dependencies dynamically across layers and positions. Specifically, each stream is functionally analogous to an addressable memory, in which attention heads can write to and read from subspaces of the main embedding space. Such information may include, for example, syntactic dependencies between tokens in the input sequence.

A common intervention method used in mechanistic interpretability research is activation patching (Zhang & Nanda 2023), also known as causal tracing (Meng et al. 2023) and interchange intervention (Geiger et al. 2021) (fig. 3).[7] The basic method involves three steps. First, the neural network must be run on an original input that relates to some behavior of interest (e.g., answering factual questions). For example, the original input might be "The capital of France is...", with the expected output being "Paris". Importantly, the activations of the network during the forward pass on this original input are cached for later use. The second step is to run the model again with an alternative input that introduces a key variation on the original input that changes the behavior (output). For example, an alternative input might be "The capital of Germany is...", with the expected output being "Berlin". Finally, the alternative input is run again but a specific component of the network's activation is swapped for its cached value from the original forward pass – an intervention known as 'patching', because it patches the alternative forward pass with a component of the original forward pass. The effect of patching a component's activation is then evaluated by comparing the model's performance in the regular forward pass on the alternative input versus the patched forward pass on the same input.

[6] In fact, it is even possible to *program* a Transformer from scratch using a specialized programming language like RASP (Restricted Access Sequence Processing Language). RASP allows mapping the basic components of a Transformer, like attention and feed-forward layers, into simple primitives that can be composed into programs. These RASP programs can then be compiled into the weights of a Transformer network that implements the specified computation (Weiss et al. 2021). More recently, methods have been developed to train Transformers whose weights are constrained to implement human-interpretable RASP-like programs. These Transformer Programs can be learned end-to-end from data and decompiled back into discrete, readable code (Friedman et al. 2023). Such approaches provide a more direct way to understand the computations of a Transformer in terms of modular algorithmic components.

[7] Note that some of these labels have been used in slightly different ways by different researchers. The most neutral term is probably "interchange intervention", as it arguably subsumes the others (Geiger et al. 2024).

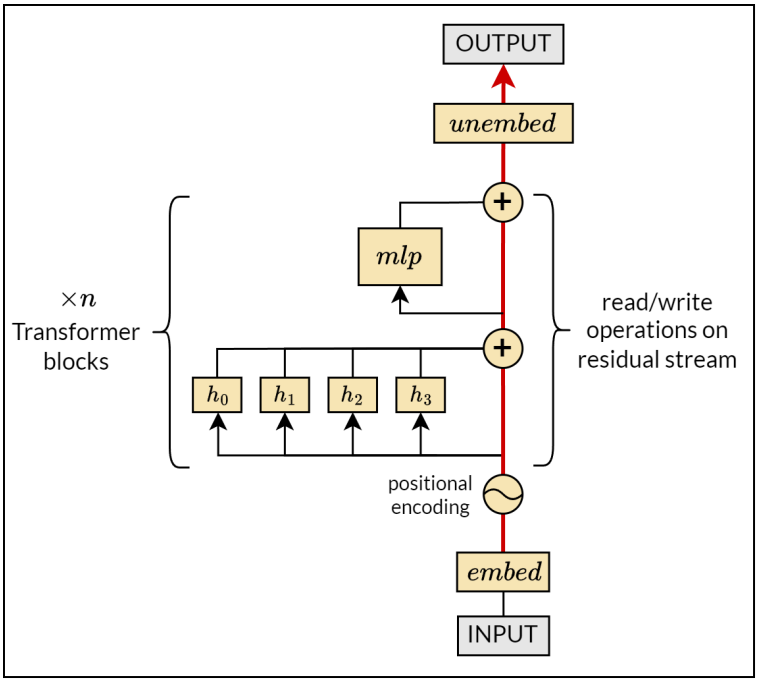

Figure 2 | **The residual stream view of the Transformer.** Each input token is first embedded into a dense vector representation and combined with a positional encoding injecting information about their position in the input sequence. This forms the initial state of the "residual stream" (depicted by the red arrow), which flows through the entire network. Each Transformer block, consisting of multi-head self-attention and a multi-layer perceptron (MLP), reads from the residual stream, transforms the representation, and writes the result back into the stream via residual connections. This process is repeated across multiple Transformer blocks. Finally, the output of the residual stream is 'unembedded' to map the transformed representation back to the original token space. In this view, the Transformer's components are seen as operators that successively refine the residual representation.

For example, the metrics used could be the probability the model assigns to the original, correct token ("Paris"), or the difference in logits[8] between the correct and incorrect tokens ("Paris" vs "Berlin"). Intuitively, changing the input hurts model performance on the expected behavior, while patching activations from the original run helps restore it. So if patching a particular component leads to a significant restoration of performance (e.g., an increase in the probability of "Paris" being the output), it suggests that component is important for the model's behavior on the task (i.e. the component not only contains information on the target feature, but that information is causally implicated in producing the desired result). By iterating this procedure over many components of the computational graph, such as attention heads in a Transformer model, activation patching aims to identify the key circuits that enable behaviors like factual recall or logical reasoning.

The mechanistic interpretability framework has been applied to many different problems, and has started shedding light on some abilities of Transformers. It is worth noting that much of this research is painstaking work conducted with toy models that are easier to interpret. While these toy models are based on the Transformer architecture, they may differ from LLMs in terms of architectural details, learning objective, dataset, and of course size (parameter count). Nonetheless, efforts are underway to automate and scale mechanistic interpretability techniques to bona fide LLMs, with promising

[8]Logits are the non-normalized model outputs taken as estimates of the probability of a particular token being the likeliest next output.

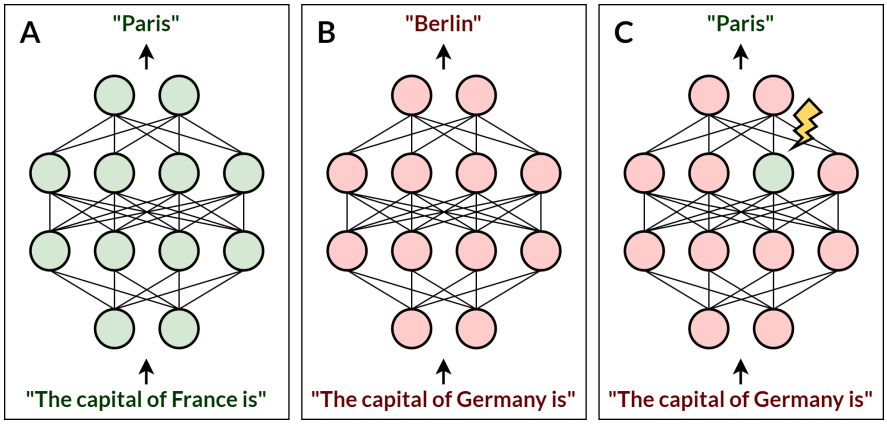

Figure 3 | **Activation patching. A.** In the original forward pass, the model takes as input the prompt "The capital of France is" and outputs the correct answer "Paris". The model activations from this forward pass are cached. **B.** In the alternative forward pass, the prompt is changed to "The capital of Germany is". The model now outputs "Berlin" as the answer. **C.** Activation patching is applied in a third forward pass. The model is given the alternative prompt "The capital of Germany is" once again, but a specific component of the model has its activations replaced (*patched*) with those from the original forward pass on the France prompt. This causes the model to output "Paris" instead of "Berlin", despite being given the Germany prompt. The restoration of the original output through patching the activations of a particular model component provides evidence that this component encodes information that is causally implicated in the target behavior. (Note that nodes in these diagrams will typically not correspond to individual neurons in a network, but rather more complex patterns or directions in activation space.)

initial results (Wu et al. 2023, Conmy et al. 2023, Syed et al. 2023). In what follows, we will briefly illustrate the fruitfulness of the mechanistic interpretability program through three case studies.

3.2.1. Case study 1: Induction heads

A canonical example of circuit discovery in language models through the tools of mechanistic interpretability is that of so-called "induction heads" (Olsson et al. 2022). These are specialized attention heads that emerge through training even in very small Transformer models, and implement a form of on-the-fly pattern completion. In other words, they allow models to repeat or generalize sequences in novel patterns that appear within prompts – potentially explaining a large amount of the so-called "in-context" learning exhibited by LLMs, where they can exhibit impressive few-shot learning of patterns not present in their training sets (Mirchandani et al. 2023).

Specifically, induction heads use a "prefix matching" attention pattern to look back over previous tokens in the current context window and detect if any match the current token. Rather than relying merely on memorized statistics about which tokens tend to follow others, they flexibly attend to whichever prior token is most similar to the current one based on learned representations. If a previous token is sufficiently similar, the induction head will attend to the next token after it. The head then increases the probability of that attended next token, effectively predicting that the current sequence will continue like the previous matched sequence. For example, if the input contains the sequence "...the cat sat on the mat. The cat...," the induction head matches the second "cat" token to the first one, attends to "sat" from the first sequence, and increases the likelihood of outputting "sat" again. This allows Transformer models to repeat sequences and generalize patterns.

Importantly, the computations performed by induction head circuits do not rely on bigram statistics memorized from the training data; rather, they operate over abstract patterns in the input sequence (prompt), even if the latter does not contain familiar strings. When the network is processing a sequence `[A][B]...[A]`, the induction head circuit can be characterized algorithmically as storing the value of the first `[A]` token in a specific subspace of the residual stream – which we may call the `previous_token` subspace – at the position of the `[B]` token. Importantly, this operation occurs whatever that specific value (content) of `[A]` might be. Functionally, the `previous_token` subspace in which this information is stored operates rather like a variable in a classical symbolic program. Its value is accessed at the next layer by the second part of the induction head circuitry, which reads the content of the `previous_token` subspace in the residual stream at the position of the `[B]` (fig. 4).[9]

3.2.2. Case study 2: Modular addition

Mechanistic interpretability is helpful to investigate not only how trained neural networks process information through algorithmic circuits, but also how they *learn* such algorithms during the course of training. Studying the training dynamics of neural networks is important, because it can provide insights into learning transition phases that are highly relevant to ongoing debates about the capacities of LLMs. Thus, Nanda et al. (2022) investigated the puzzling phenomenon of grokking, where neural networks trained on algorithmic tasks with regularization initially overfit to the training data but later suddenly generalize after many training steps. As a case study, the authors trained small Transformer models on modular addition tasks. They find these networks exhibit "grokking", initially overfitting but later learning to generalize.

To understand this phenomenon, they reverse engineered the mechanisms learned by these networks using techniques from mechanistic interpretability. That found that the network learns to perform modular addition tasks by mapping inputs onto rotations in the plane and composing those rotations using trigonometric identities (fig. 5). This clever algorithm, dubbed "Fourier multiplication", allows the network to perform addition given a particular prime number chosen as a modulus (where the modulus determines the size of the "circle" around which the addition operation travels–see Fig. 5 below).

Using this understanding, Nanda et al. defined new progress metrics allowing them to study the training dynamics of the models as they learn to perform modular addition algorithmically. They identified three distinct learning phases, each marked by continuous progress on certain metrics:

1. *Memorization phase*: In the early part of training, the models fit to the training data by simply memorizing input-output pairs. Performance on the test set remains low while performance on the training set increases rapidly, indicating the models are overfitting.
2. *Circuit formation phase*: After memorization, there is a transition period where the models internalize the algorithm for modular addition using trigonometric identities and rotations (the "Fourier multiplication" circuit). However, performance on the test set remains low, implying memorization components still persist.
3. *Cleanup phase* Finally, weight decay drives the removal of the initial memorization components. Performance on the test set abruptly improves to match performance on the training set at the end of this phase, corresponding to the "grokking" transition in generalization capability.

The entire phenomenon is thus not a sudden onset of generalization ability, but rather a gradual amplification of structured mechanisms encoded in the weights, followed by pruning of unnecessary

[9] For a discussion of induction heads circuitry as implementing a form of variable binding, and the implications of this view for the debate about compositionality in connectionist models, see Millière (2024).

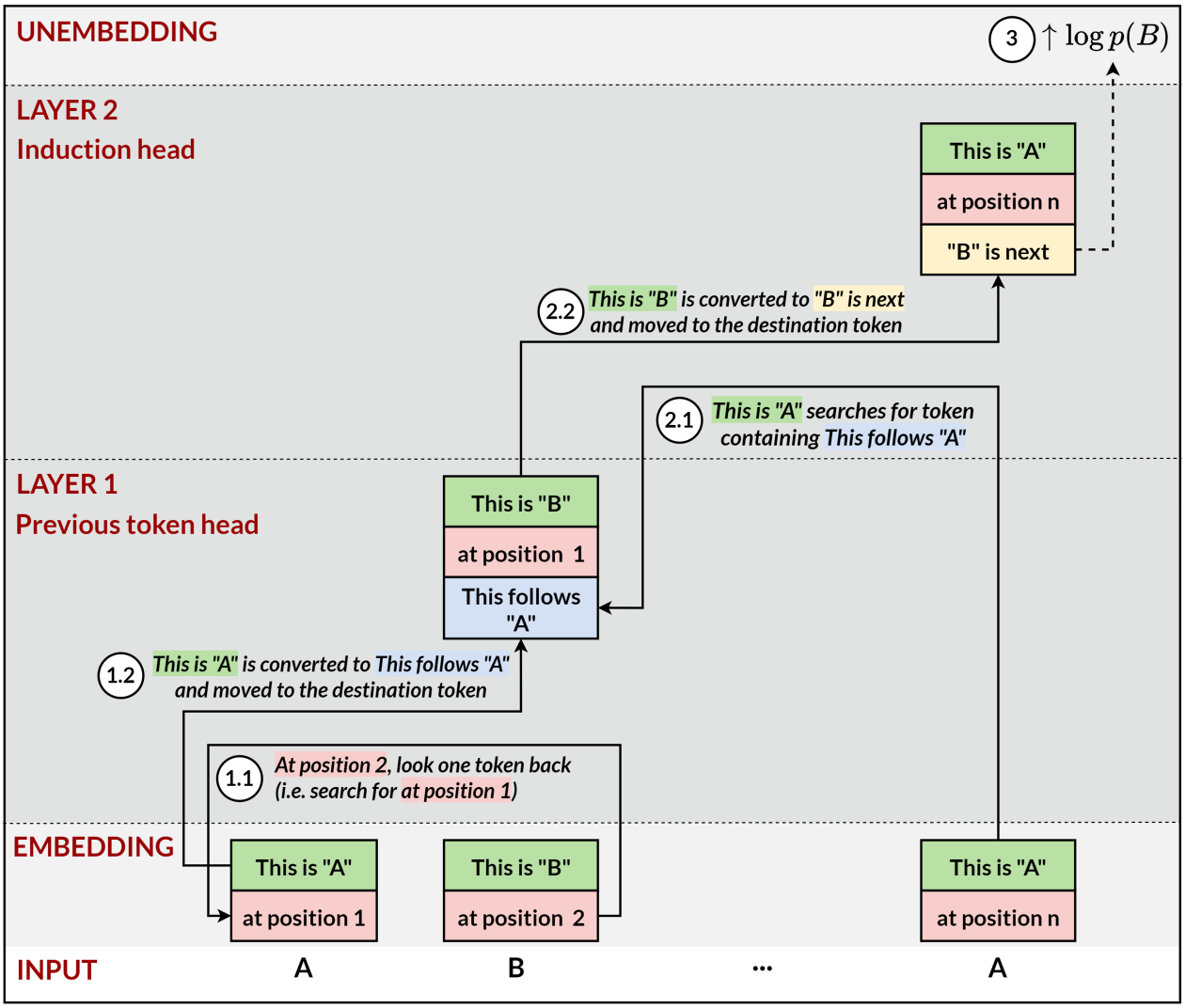

Figure 4 | **A schematic illustration of the induction head circuit in a two-layer Transformer model.** At the embedding stage, each token from the input sequence is encoded as a vector, together with information about its position in the sequence. The first layer contains an attention head – known as the *previous token head* – that acquired a specialized function during training. When processing token `[B]` at position 2, the previous token head does the following: (1.1) it attends to the previous token at position 1; (1.2) It writes the identity of this preceding token to a dedicated subspace of the residual stream at the current position (position 2), effectively storing the information "the token before me is `[A]`". Layer 2 contains another specialized attention head known as the *induction head*. When processing the second instance of `[A]` at position n, the induction head does the following: (2.1) it queries the residual stream for information in the "previous token" subspace matching the current token's identity; (2.2) having located this previous token information in the residual stream at position 2, it retrieves the identity of the token at that position (`[B]`), then writes this identity to a dedicated subspace of the residual stream at the current position (position n), effectively storing the information "predict that the next token will be `[B]`". (3) The unembedding layer maps information in the "next token" subspace at position n to an increased logit for `[B]` at position $n+1$, which translates to an increased log likelihood of `[B]` being predicted as the next token.

components. Importantly, these findings speak against the skeptical view of LLMs discussed in Section

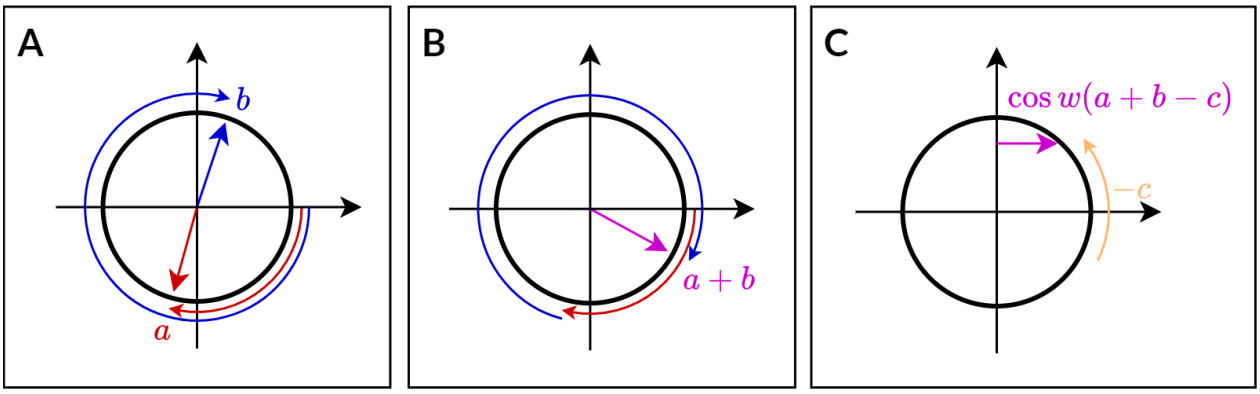

Figure 5 | **A learned algorithm for modular addition** (figure adapted from Nanda et al. (2022)). **A. Embedding projection.** Given two input numbers a and b in the modular addition $a + b \equiv c \mod P$, the model uses its embedding matrix to project each number onto a corresponding rotation around the unit circle. The embedding matrix essentially memorizes a mapping between each possible input number and a specific rotation amount, converting the numbers into geometric representations. **B. Rotation composition.** The model composes the two rotations generated for a and b. This step effectively adds the two rotation amounts together, resulting in a new, single rotation that represents the sum $a + b$ in modular arithmetic. In modular arithmetic, numbers "wrap around" after exceeding the modulus P, so if $a + b$ is greater than P, the resulting rotation will correspond to $a + b \mod P$, which is the remainder when $a + b$ is divided by P. **C. Output decoding.** To produce the output logits (raw scores used for next-token prediction), the model considers each possible result c (ranging from 0 to $P - 1$) and performs a reverse rotation by $-c$. This step essentially checks, for each c, whether undoing the rotation by c results in a rotation that matches the one representing $a + b \mod P$. The output c that produces the rotation most closely matching the $a + b \mod P$ rotation is assigned the highest logit. This works because the trained model ensures that the correct c satisfying $a + b \equiv c \mod P$ will undo the rotation by exactly the right amount to point back to the $a + b \mod P$ rotation. The trigonometric functions cosine and sine are used to implement these rotations and achieve the desired result mathematically using angle addition identities, but conceptually, the algorithm is based on representing numbers as rotations and composing these rotations together.

1, according to which their abilities can be explained by mere memorization. While memorization dominates in the initial phase of training, subsequent phase transitions show that Transformer models can learn general, rule-like algorithms to solve tasks, including tasks as rigid and well-defined as arithmetic problems.

3.2.3. Case study 3: World models

There is an ongoing debate about whether LLMs may acquire "world models," characterized as sets of structure-preserving internal representations that encode abstract information about real-world or domain-specific entities, relations, and processes – rather than merely intralinguistic features (Yildirim & Paul 2023, Mollo & Millière 2023). Inducing such representations could allow LLMs to generate outputs consistent with real-world constraints, reason about worldly phenomena, and potentially misrepresent aspects of reality, despite lacking direct sensorimotor experience. While behavioral evaluations are generally insufficient to settle this debate, mechanistic interpretability shows promise to uncover what the LLMs and related Transformer models represent internally.

To investigate whether Transformer models trained to predict sequences can learn interpretable world representations (rather than mere surface statistics), Li et al. (2023) focus on the board

game Othello as a simplified yet non-trivial domain. They trained a GPT variant (Othello-GPT) to sequentially predict tokens representing Othello board positions. The only inputs to the model are move sequences derived from game transcript; no explicit game rules or board structure are provided. After training on championship games and synthetic games, Othello-GPT recommends legal moves with high accuracy, suggesting it has learned more than surface statistics. Furthermore, nonlinear probes reliably predict board states from model activations, implying that a nonlinear representation of the board state had emerged during training. To validate the probes' accuracy, the authors performed intervention experiments that modify Othello-GPT's internal activations to reflect altered board states. The model's subsequent move predictions change accordingly, confirming the causal role of these latent board state representations.

In a follow-up study, Nanda et al. (2023) discovered that rather than representing the board state (e.g., representing each board tile as `black`, `white`, or `empty`), Othello-GPT actually encodes tiles relative to the current player (as `player`, `opponent`, or `empty`). By re-orienting probes to classify this player-centric representation, Nanda et al. demonstrated that the board state is in fact *linearly* encoded with high accuracy in the network, contrary to Li et al. (2023)'s claim that the board state is only encoded non-linearly (fig. 6). They further demonstrated behavioral control by conducting simple vector arithmetic interventions to alter the model's encoding of board states and change predictions accordingly. Hazineh et al. (2023) found similar evidence that information about board state is encoded in a simple, linear way in the deeper layers of Othello-GPT models. Like Nanda et al. (2023), they decoded a representation corresponding to tiles marked `player`, `opponent`, or `empty`, which aligns well with the role of the model in alternating between playing as white or black. To test whether these internal representations play a causal role in the model's predictions, they also intervened by manipulating activations to trick the model about the state of the board. Through visualizing effects on predicted logits and comparing distributional similarity of logit outputs, they demonstrated layers in which this internal representation steers next-move predictions. The internal representation appears fully developed and utilized in middle layers of deeper models, while shallow models fail to use the representation causally.

Studies on Othello-GPT provide tentative evidence that Transformer models can acquire world models at least in toy domains.[10] While Othello-GPT is not a language model properly speaking, it generates legal game moves in written form, and interventionist experiments reveal that it generates such moves on the basis of linearly decodable representations of the board state. To what extent can we generalize these findings to actual LLMs? This is a challenging question. A reasonable assumption is that Othello-GPT acquires a world model – an internal representation of the game world – because this is useful to improve its predictions of legal moves past a certain threshold. However, neural networks are notorious in their ability to achieve high accuracy scores on problems by mastering surface statistics; they tend to learn shortcuts (e.g., shallow heuristics) to achieve good performance on their learning objective until they hit a bottleneck. Furthermore, acquiring world models about the real world, rather than an extremely simple game world, is presumably costly in terms of training dynamics; it would require a substantial reorganization of the model's internal structure, akin to the phase transition undergone by Nanda et al. (2022)'s model to "grok" modular addition, albeit on a much broader scale.

It may well be that LLMs do undergo such phase transitions during training, and acquire representations akin to world models at least in some limited domains. This might be required to unlock certain abilities such as commonsense reasoning about intuitive physics, which in turn would further reduce the loss (i.e., improve next-token prediction performance) in certain contexts. For this to happen, two conditions must presumably be satisfied: (a) the text-based training data should provide enough

[10] For example, similar results have been found by studying Chess-GPT, a Transformer model trained on chess moves (Karvonen 2024).

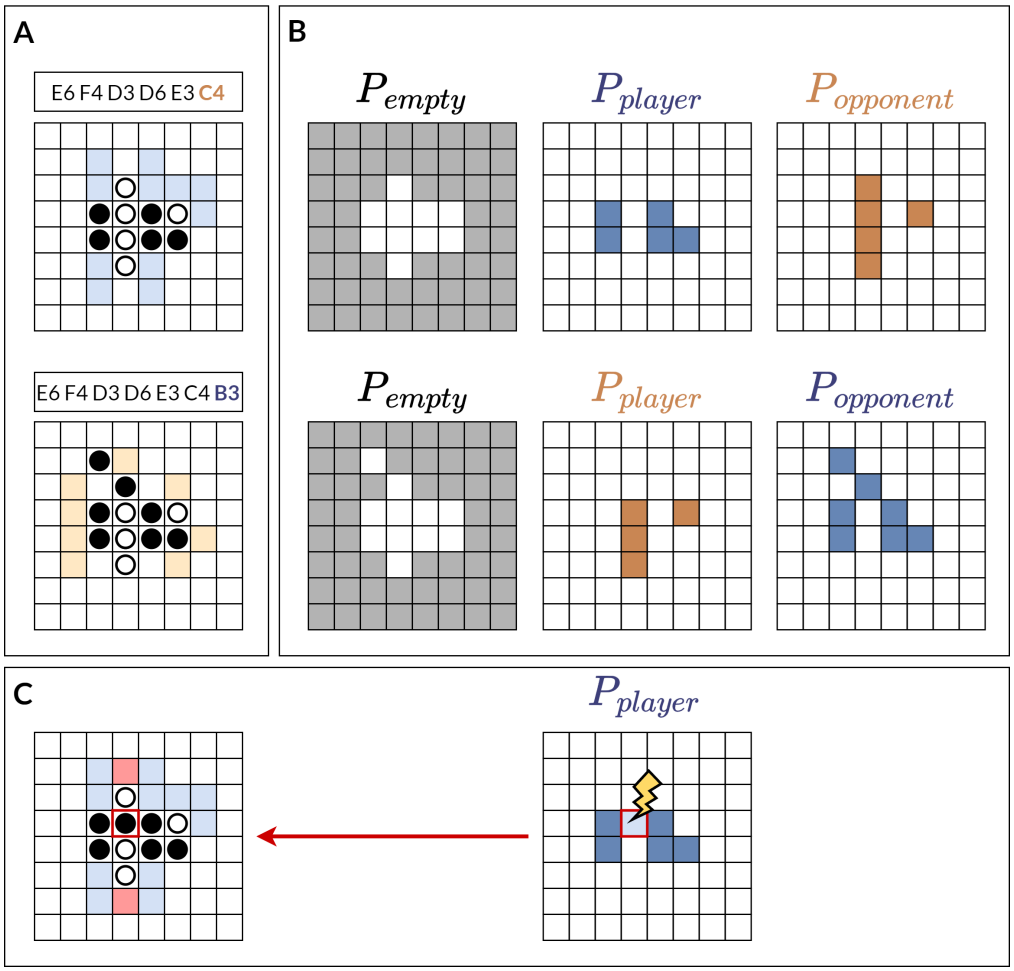

Figure 6 | **Emergent representations of the board state in Othello-GPT** (adapted from Nanda et al. (2023)). The model represents board states relative to the current player. **A.** The ground truth board states at two consecutive time steps. Colored tiles show legal moves for the current player (light blue for BLACK and light orange for WHITE). **B.** The board states at the same two time steps decoded by linear probes. The probes are trained to classify the board relative to the current player: `empty` for empty tiles, `player` for tiles occupied by the current player, and `opponent` for tiles occupied by the current player's opponent. Note that the `player` and `opponent` colors are flipped between the two time steps, as the current player changes. **C.** We can intervene on the model's internal board state representation by pushing an empty tile's representation in the direction of the `player` vector, to make the model represent that tile as occupied by the current player. This simple linear intervention is sufficient to alter Othello-GPT's move predictions (red tiles on the left), demonstrating that the linearly represented board state causally determines the model's outputs.

information to induce the relevant components of world models; (b) the learning pressure to reduce loss on the next-token prediction objective should be sufficient to push the model to acquire such representations given long enough training trajectories. Neither condition is antecedently guaranteed, and there is currently limited evidence that they apply to most existing LLMs. In particular, interventionist evidence regarding the most capable LLMs such as GPT-4, whose behavior on certain tasks is most consistent with the acquisition of world models, is still lacking. As far as behavioral evidence is concerned, lingering failure modes on out-of-distribution tasks (those with an informational structure different from the training set, for example involving the same physical principles but applied to

different objects or in different contexts) cast doubt on the hypothesis that current LLMs acquire world models that are as robust as those discovered by humans and animals (McCoy et al. 2023, Yildirim & Paul 2023).

There are nonetheless ongoing efforts to assess the existence of world models in LLMs using the tools of mechanistic interpretability. For example, Li et al. (2021) used probing and intervention methods on models fine-tuned on the Alchemy and TextWorld datasets. Alchemy contains sequences of instructions for manipulating colored liquids in beakers (i.e., instructions for performing fictional chemical experiments and their outcomes), while TextWorld consists of textual transcripts of navigation in simulated worlds. Li et al. designed probes to test if the models' contextual token embeddings encode and track the state of entities mentioned in the discourse. For example, in Alchemy the probe tries to determine if a representation encodes that a beaker is empty after its contents are drained. Li et al. used these probes to test whether intervening on the decoded entity representations would change model behavior. Specifically, they constructed two discourses x_1 and x_2 that describe draining liquid from two different beakers, b_1 and b_2, resulting in one empty beaker per discourse. After encoding each discourse, they created a synthetic representation C_{mix} by taking the encoding C_1 of discourse x_1 and replacing the vector representations corresponding to the initial description of beaker b_2 with those from C_2. Although C_{mix} does not correspond to any real textual input, it implicitly represents a situation in which both b_1 and b_2 are empty. When generating text conditioned on C_{mix}, they found that the model generates instructions that are more often consistent with both beakers being empty compared to generating from C_1 or C_2 alone. However, the generated instructions from C_{mix} are still not always fully consistent with the implicit state, suggesting the induced representation is approximate rather than perfect. By editing the entity representations to model a new state not seen in the actual training data or prompt, and observing changes in the model's outputs that are often consistent with this new state, Li et al. provided tentative evidence that the model can induce an approximate implicit representation of the state of discourse entities purely from text. Note that producing text that is consistent with the hypothesis-driven manipulation of the LLMs activation space is not something that the system was trained to do. Furthermore, the method was found to be robust against "sanity checks" to confirm that the intervention itself is not wholly responsible for injecting appropriate information into the system, such as finding negative results when attempting the intervention method on similarly-complex Transformer architecture with randomized weights. However, the generality of these findings remains uncertain, as the experiments focused on narrow domains with simple objects and dynamics. More systematic testing would be needed to determine how well they generalize to more complex and open-ended settings.

3.3. Dictionary learning with sparse autoencoders

While early research in mechanistic interpretability has largely focused on toy models, where probing experiments and causal interventions are more tractable, recent work has explored a new method called *dictionary learning* which aims to automatically identify interpretable features in LLMs. This approach seeks to address the challenge of superposition, whereby neural networks represent more features than they have neurons by encoding each feature as a combination of multiple neurons' activations.

Dictionary learning uses sparse autoencoders (SAEs) to decompose the dense activation patterns of LLMs into sparse combinations of interpretable features. SAEs are simple unsupervised neural networks that learn to reconstruct their input data using a large hidden layer with many more neurons than the input, while enforcing sparsity constraints to ensure that only a small number of hidden neurons activate for any given input. The process of dictionary learning involves training an SAE on activation vectors from a specific layer of the LLM. The autoencoder learns to reconstruct these

activations using a larger set of sparse features. Sparsity constraints are applied during training to encourage each feature to represent a single, interpretable concept. The result is a "dictionary" of features that ideally correspond to human-understandable concepts.

Bricken et al. (2023) established the effectiveness of this approach by showing that SAEs can extract features representing specific contexts, such as Arabic script and DNA sequences, as well as abstract concepts like code errors and mathematical operations from language models. Building on this work, Templeton et al. (2024) and Gao et al. (2024) further demonstrated that this approach scales to state-of-the-art LLMs like Claude and GPT-4, successfully extracting millions of sparse features spanning a wide range of concepts. For example, a sparse "Golden Gate Bridge" feature in Claude activates strongly on both images and text inputs – across multiple languages – related to the Golden Gate Bridge (Templeton et al. 2024).

This approach offers several advantages over older methods like probing. It is highly scalable, efficiently producing millions of reusable features. It can be used to automate the discovery of unanticipated features without relying on hand-crafted probes, which potentially allows for a more comprehensive exploration of the model's internal representations. The extracted sparse features often seem to map onto human-understandable concepts, much reliably than the activations of individual neurons in the original model. Once extracted, these interpretable sparse features can be reused for various analyses and interventions.

Despite the excitement surrounding dictionary learning, however, it still lacks robust theoretical grounding compared to more established intervention techniques in mechanistic interpretability. There is a notable absence of well-established metrics for evaluating the quality of SAE-extracted features and the overall performance of the SAE. Without a consensus on such metrics, it is difficult to objectively assess whether a given dictionary learning experiment has succeeded in extracting sparse features that are actually involved in the LLMs' computations. The allure of "autointerpretability" – using LLMs themselves to automatically label SAE-extracted features – introduces additional methodological concerns. This approach may create an illusion of understanding by latching onto spurious correlations rather than identifying genuine causal relationships. LLM-generated labels may seem to make sense at first glance based on a subset of highly activating samples for a given sparse feature, but can actually be misleading on closer inspection. Addressing these limitations on firm theoretical and methodological grounding will be important to ensure that dictionary learning actually provides reliable, meaningful insights into the inner workings of LLMs.

3.4. Interpretability and causal abstraction

The general project of mechanistic interpretability can be seen through the lens of causal abstraction developed by (Geiger et al. 2021). Causal abstraction is a theoretical framework that aims to provide an interpretable high-level causal explanation of the behavior of a complex system, such as a neural network, that is consistent with the low-level causal structure of that system.

The core intuition is that the variables of a large, complex causal model can be clustered into sets and aligned with the variables of a sparse, low-dimensional model with fewer, high-level variables. The high-level model provides an abstract characterization of the low-level model if aligned high-level and low-level variables have equivalent causal roles and information content. This equivalence is experimentally verified with interventions on both levels that produce the same counterfactual behavior. More formally, an alignment between a low-level model $\mathcal{L}$ and high-level model $\mathcal{H}$ is a partition of the low-level variables into sets, with each set aligned to a high-level variable. A translation function maps low-level variable values to high-level values (Geiger et al. 2024). The alignment is causally consistent if, for any low-level intervention that has a corresponding high-level

intervention under the translation function, intervening on both models produces equivalent results after translation. If an alignment is causally consistent, then $\mathcal{H}$ is said to be a constructive causal abstraction of $\mathcal{L}$.

Any program or algorithm can be represented as a causal model (Icard 2017). In the context of interpretability research in deep learning, the low-level causal model is a neural network, containing many interconnected nodes and weights that determine its function. The high-level causal model proposed by mechanistic interpretability researchers offer hypotheses about the abstract computations and algorithms implemented by the network. Aligning groups of neurons and weights to single variables in the high-level model allows interpreting their collective causal role. The causal consistency condition ensures the high-level causal model faithfully captures the causal mechanisms embodied by the low-level neural network model. Causal abstraction thus enables the development of human-interpretable high-level causal models that accurately explain the reasoning and computations inside opaque neural networks.

A key method for assessing causal abstraction involves "interchange interventions", where neural representations created for one input are swapped into the model when it is processing another input. If the low-level neural model and high-level algorithm have the same counterfactual behavior under aligned interchange interventions, that provides evidence that the alignment represents and exploits a causal abstraction. Many of the methods described above, such as activation patching, can be used as interchange interventions. In fact, iterative nullspace projection can also be formalized within a causal abstraction framework (Geiger et al. 2024). Thus, causal abstraction is a useful framework to unify interventionist approaches to interpretability.

In practice, the project of mechanistic interpretability with large neural networks like LLMs can be seen as aiming for *approximate* causal abstraction, which relaxes the criteria for alignment (Beckers et al. 2020). The degree of abstraction can be quantified as the proportion of aligned interchange interventions that have equivalent effects in the high-level causal model and low-level causal model or target neural network (Geiger et al. 2024). When interchange intervention accuracy is 100%, the high-level model exactly abstracts the neural network. Otherwise, the high-level model approximately abstracts the neural network approximately, to the degree quantified by interchange intervention accuracy. Approximate abstraction can provide an interpretable high-level explanation that still reflects the causal structure of the target neural network with a degree of faithfulness suitable for genuine explanation.

4. Conclusion

In this paper, we motivated and reviewed recent interventionist approaches to interpreting the inner workings of deep neural networks, with a particular focus on Transformer-based language models. We emphasized that purely behavioral methods have significant limitations and argued that mechanistic approaches to interpretability, which aim to elucidate the causal structure of a model's internal organization, offer a more promising avenue from a philosophy of science perspective. To illustrate these points, we presented a series of case studies showcasing recent interpretability techniques. These examples demonstrate the potential of interventionist methods to provide insights into the complex operations of neural networks that were previously considered opaque. We concluded by reviewing recent attempts to develop a general theory of causal abstraction for neural networks; this emerging theoretical framework has the potential to enable more ambitious and systematic applications of interventionist methods. Our analysis challenges the prevalent view that deep neural networks are inscrutable black boxes whose inner workings defy comprehension. Notably, techniques

such as activation patching or interchange interventions offer a degree of control over the underlying system that neuroscientists cannot currently aspire to achieve with biological neural networks.

It may be helpful to reiterate some limitations of these methods in closing. For one, while these methods offer potential insights into the causal structure of complex large language models, they typically require full access to their internal weights. Given the proprietary nature of state-of-the-art LLMs like GPT-4 and Claude, academics and regulatory agencies without internal access may find limited practical applications for these methods in addressing scientific questions or trust and safety concerns. Additionally, many of these methods are computationally intensive. For instance, researchers at Anthropic have noted that systematic dictionary learning experiments with Claude would require more computational power than the initial training of the model itself (Templeton et al. 2024). Lastly, as with all emerging scientific methods, these techniques will likely present their own unique methodological challenges, including interpretability illusions resulting from spurious features or causal dependencies. We review these methods here in hope that it will facilitate future research to address the ethical, practical, and scientific concerns associated with these novel approaches.

References

Aiyappa, R., An, J., Kwak, H. & Ahn, Y.-Y. (2023), 'Can we trust the evaluation on ChatGPT?'.

Alain, G. & Bengio, Y. (2018), 'Understanding intermediate layers using linear classifier probes'.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J. H., Shafey, L. E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G. H., Ahn, J., Austin, J., Barham, P., Botha, J., Bradbury, J., Brahma, S., Brooks, K., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C. A., Chowdhery, A., Crepy, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., Díaz, M., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., Garcia, X., Gehrmann, S., Gonzalez, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M., Li, W., Li, Y., Li, J., Lim, H., Lin, H., Liu, Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, A., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A. C., Roy, A., Saeta, B., Samuel, R., Shelby, R., Slone, A., Smilkov, D., So, D. R., Sohn, D., Tokumine, S., Valter, D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L., Yin, P., Yu, J., Zhang, Q., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S. & Wu, Y. (2023), 'PaLM 2 Technical Report'.

Bansal, H., Gopalakrishnan, K., Dingliwal, S., Bodapati, S., Kirchhoff, K. & Roth, D. (2023), Rethinking the Role of Scale for In-Context Learning: An Interpretability-based Case Study at 66 Billion Scale, *in* A. Rogers, J. Boyd-Graber & N. Okazaki, eds, 'Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)', Association for Computational Linguistics, Toronto, Canada, pp. 11833–11856.

Barwich, A.-S. (2019), 'The Value of Failure in Science: The Story of Grandmother Cells in Neuroscience', *Frontiers in Neuroscience* **13**.

Beckers, S., Eberhardt, F. & Halpern, J. Y. (2020), Approximate Causal Abstractions, *in* 'Proceedings of The 35th Uncertainty in Artificial Intelligence Conference', PMLR, pp. 606–615.

Belinkov, Y. (2022), 'Probing Classifiers: Promises, Shortcomings, and Advances', *Computational Linguistics* **48**(1), 207–219.

Bian, Y., Huang, J., Cai, X., Yuan, J. & Church, K. (2021), On attention redundancy: A comprehensive study, *in* K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty & Y. Zhou, eds, 'Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies', Association for Computational Linguistics, Online, pp. 930–945.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T. & Olah, C. (2023), 'Towards monosemanticity: Decomposing language models with dictionary learning', *Transformer Circuits Thread* .

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T. & Zhang, Y. (2023), 'Sparks of Artificial General Intelligence: Early experiments with GPT-4'.

Buckner, C. (2019), 'Deep learning: A philosophical introduction', *Philosophy Compass* **14**(10), e12625.

Chefer, H., Gur, S. & Wolf, L. (2021), Transformer Interpretability Beyond Attention Visualization, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 782–791.

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S. & Garriga-Alonso, A. (2023), 'Towards Automated Circuit Discovery for Mechanistic Interpretability'.

Craver, C. F. (2007), *Explaining the Brain: Mechanisms and the Mosaic Unity of Neuroscience*, Oxford University Press, Clarendon Press, New York : Oxford University Press,.

Doerig, A., Sommers, R. P., Seeliger, K., Richards, B., Ismael, J., Lindsay, G. W., Kording, K. P., Konkle, T., van Gerven, M. A. J., Kriegeskorte, N. & Kietzmann, T. C. (2023), 'The neuroconnectionist research programme', *Nature Reviews Neuroscience* **24**(7), 431–450.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M. & Olah, C. (2022), 'Toy models of superposition', *Transformer Circuits Thread* .

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S. & Olah, C. (2021), 'A mathematical framework for transformer circuits', *Transformer Circuits Thread* .

Friedman, D., Wettig, A. & Chen, D. (2023), 'Learning Transformer Programs'.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J. & Wu, J. (2024), 'Scaling and evaluating sparse autoencoders'.

Geiger, A., Ibeling, D., Zur, A., Chaudhary, M., Chauhan, S., Huang, J., Arora, A., Wu, Z., Goodman, N., Potts, C. & Icard, T. (2024), 'Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability'.

Geiger, A., Lu, H., Icard, T. & Potts, C. (2021), Causal Abstractions of Neural Networks, *in* 'Advances in Neural Information Processing Systems', Vol. 34, Curran Associates, Inc., pp. 9574–9586.

Gromov, A., Tirumala, K., Shapourian, H., Glorioso, P. & Roberts, D. A. (2024), 'The Unreasonable Ineffectiveness of the Deeper Layers'.

Grynbaum, M. M. & Mac, R. (2023), 'The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work', *The New York Times* .

Harding, J. (2023), 'Operationalising Representation in Natural Language Processing'.

Hazineh, D. S., Zhang, Z. & Chiu, J. (2023), 'Linear Latent World Models in Simple Transformers: A Case Study on Othello-GPT'.

He, S., Sun, G., Shen, Z. & Li, A. (2024), 'What Matters in Transformers? Not All Attention is Needed'.

Hewitt, J. & Liang, P. (2019), Designing and Interpreting Probes with Control Tasks, *in* 'Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)', Association for Computational Linguistics, Hong Kong, China, pp. 2733–2743.

Hupkes, D., Veldhoen, S. & Zuidema, W. (2018), 'Visualisation and 'Diagnostic Classifiers' Reveal How Recurrent and Recursive Neural Networks Process Hierarchical Structure', *Journal of Artificial Intelligence Research* **61**, 907–926.

Icard, T. F. (2017), From programs to causal models, *in* 'Proceedings of the 21st Amsterdam Colloquium', pp. 35–44.

Jonas, E. & Kording, K. P. (2017), 'Could a Neuroscientist Understand a Microprocessor?', *PLOS Computational Biology* **13**(1), e1005268.

Karhade, M. (2023), 'GPT-4: 8 Models in One ; The Secret is Out'.

Karvonen, A. (2024), 'Chess-GPT's Internal World Model', https://adamkarvonen.github.io/machine_learning/2024/
world-models.html.

Li, B. Z., Nye, M. & Andreas, J. (2021), Implicit Representations of Meaning in Neural Language Models, *in* 'Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)', Association for Computational Linguistics, Online, pp. 1813–1827.

Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H. & Wattenberg, M. (2023), 'Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task'.

Lindsay, G. W. & Bau, D. (2023), 'Testing methods of neural systems understanding', *Cognitive Systems Research* **82**, 101156.

Lipton, Z. C. (2018), 'The mythos of model interpretability', *Communications of the ACM* **61**(10), 36–43.

Machamer, P., Darden, L. & Craver, C. F. (2000), 'Thinking about Mechanisms', *Philosophy of Science* **67**(1), 1–25.

Marr, D. (1982), *Vision: A Computational Approach*, Freeman & Co.

McCoy, R. T., Yao, S., Friedman, D., Hardy, M. & Griffiths, T. L. (2023), 'Embers of Autoregression: Understanding Large Language Models Through the Problem They are Trained to Solve'.

Meng, K., Bau, D., Andonian, A. & Belinkov, Y. (2023), 'Locating and Editing Factual Associations in GPT'.

Meyes, R., Lu, M., de Puiseau, C. W. & Meisen, T. (2019), 'Ablation Studies in Artificial Neural Networks'.

Millière, R. (2024), 'Philosophy of cognitive science in the age of deep learning', *WIREs Cognitive Science* **n/a**(n/a), e1684.

Millière, R. & Buckner, C. (2024), 'A Philosophical Introduction to Language Models – Part I: Continuity With Classic Debates'.

Mirchandani, S., Xia, F., Florence, P., Ichter, B., Driess, D., Arenas, M. G., Rao, K., Sadigh, D. & Zeng, A. (2023), 'Large Language Models as General Pattern Machines'.

Mollo, D. C. & Millière, R. (2023), 'The Vector Grounding Problem'.

Nanda, N., Chan, L., Lieberum, T., Smith, J. & Steinhardt, J. (2022), Progress measures for grokking via mechanistic interpretability, *in* 'The Eleventh International Conference on Learning Representations'.

Nanda, N., Lee, A. & Wattenberg, M. (2023), 'Emergent Linear Representations in World Models of Self-Supervised Sequence Models'.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S. & Olah, C. (2022), 'In-context learning and induction heads', *Transformer Circuits Thread* .

OpenAI (2023), 'GPT-4 Technical Report'.

Piccinini, G. (2020), *Neurocognitive Mechanisms: Explaining Biological Cognition*, Oxford University Press.

Plaut, D. C. & McClelland, J. L. (2010), 'Locating object knowledge in the brain: Comment on Bowers's (2009) attempt to revive the grandmother cell hypothesis', *Psychological Review* **117**(1), 284–288.

Ravfogel, S., Elazar, Y., Gonen, H., Twiton, M. & Goldberg, Y. (2020), 'Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection'.

Ravfogel, S., Prasad, G., Linzen, T. & Goldberg, Y. (2021), Counterfactual Interventions Reveal the Causal Effect of Relative Clause Representations on Agreement Prediction, *in* 'Proceedings of the 25th Conference on Computational Natural Language Learning', Association for Computational Linguistics, Online, pp. 194–209.

Räz, T. & Beisbart, C. (2022), 'The Importance of Understanding Deep Learning', *Erkenntnis* .

Rumelhart, D. E., Mcclelland, J. L. & Group, P. R. (1987), *Parallel Distributed Processing, Volume 1: Explorations in the Microstructure of Cognition: Foundations*, MIT Press.

Sejnowski, T. J. & Rosenberg, C. R. (1987), 'Parallel Networks that Learn to Pronounce English Text', *Complex System* **1**, 145–168.

Smolensky, P. (1986), Neural and conceptual interpretation of PDP models, *in* 'Parallel Distributed Processing: Explorations in the Microstructure, Vol. 2: Psychological and Biological Models', MIT Press, Cambridge, MA, USA, pp. 390–431.

Smolensky, P. (1988), 'On the proper treatment of connectionism', *Behavioral and Brain Sciences* **11**(1), 1–23.

Syed, A., Rager, C. & Conmy, A. (2023), 'Attribution Patching Outperforms Automated Circuit Discovery'.

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C. & Henighan, T. (2024), 'Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet', *Transformer Circuits Thread* .

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S. & Scialom, T. (2023), 'Llama 2: Open Foundation and Fine-Tuned Chat Models'.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R. & Titov, I. (2019), 'Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned'.

Weiss, G., Goldberg, Y. & Yahav, E. (2021), Thinking Like Transformers, *in* 'Proceedings of the 38th International Conference on Machine Learning', PMLR, pp. 11080–11090.

Woodward, J. (2005), *Making Things Happen: A Theory of Causal Explanation*, Oxford University Press, USA.

Wu, Z., Geiger, A., Potts, C. & Goodman, N. D. (2023), 'Interpretability at Scale: Identifying Causal Mechanisms in Alpaca'.

Yildirim, I. & Paul, L. A. (2023), 'From task structures to world models: What do LLMs know?', *Trends in Cognitive Sciences* .

Yousefi, S., Betthauser, L., Hasanbeig, H., Millière, R. & Momennejad, I. (2024), 'Decoding In-Context Learning: Neuroscience-inspired Analysis of Representations in Large Language Models'.

Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. (2021), 'Understanding deep learning (still) requires rethinking generalization', *Communications of the ACM* **64**(3), 107–115.

Zhang, F. & Nanda, N. (2023), 'Towards Best Practices of Activation Patching in Language Models: Metrics and Methods'.