
The Cognitive Revolution in Interpretability: From Explaining Behavior to Interpreting Representations and Algorithms

Adam Davies

Department of Computer Science
University of Illinois Urbana-Champaign
adavies4@illinois.edu

Ashkan Khakzar

Department of Engineering Science
University of Oxford
ashkan.khakzar@eng.ox.ac.uk

Abstract

Artificial neural networks have long been understood as “black boxes”: though we know their computation graphs and learned parameters, the knowledge encoded by these weights and functions they perform are not inherently interpretable. As such, from the early days of deep learning, there have been efforts to explain these models’ behavior and understand them internally; and recently, *mechanistic interpretability* (MI) has emerged as a distinct research area studying the features and implicit algorithms learned by foundation models such as large language models. In this work, we aim to ground MI in the context of cognitive science, which has long struggled with analogous questions in studying and explaining the behavior of “black box” intelligent systems like the human brain. We leverage several important ideas and developments in the history of cognitive science to disentangle divergent objectives in MI and indicate a clear path forward. First, we argue that current methods are ripe to facilitate a transition in deep learning interpretation echoing the “cognitive revolution” in 20th-century psychology that shifted the study of human psychology from pure behaviorism toward mental representations and processing. Second, we propose a taxonomy mirroring key parallels in computational neuroscience to describe two broad categories of MI research, *semantic interpretation* (what latent representations are learned and used) and *algorithmic interpretation* (what operations are performed over representations) to elucidate their divergent goals and objects of study. Finally, we elaborate the parallels and distinctions between various approaches in both categories, analyze the respective strengths and weaknesses of representative works, clarify underlying assumptions, outline key challenges, and discuss the possibility of unifying these modes of interpretation under a common framework.

1 Introduction

How can we understand, interpret, and explain the behavior of complex intelligent biological systems like humans, chimpanzees, dolphins, or (at least seemingly) intelligent AI systems like large language models (LLMs)? This question lies at the heart of cognitive science, and different answers have defined entire paradigms in member disciplines such as psychology, neuroscience, and linguistics. For instance, in the 1950s and early 1960s, the dominant paradigm in academic psychology was behaviorism [1], resting on the fundamental assumption that human behavior can be fully understood and explained in terms of stimulus-and-response mechanisms [2, 3], with no need to consider humans’ internal mental states, representations, or processing. However, behaviorism failed to account for many key facets of human psychology, including language acquisition [3, 4], concept

learning and problem-solving [5], and working memory [6], leading to the so-called “cognitive revolution” of the 1960s that birthed the modern cognitive sciences built around studying human cognition on the basis of mental representation and information processing [7, 8].

Most contemporary deep learning research also rests upon the behaviorist assumption – i.e., that it is only necessary to study models’ behaviors (and not their internal representations, processing, etc.) in order to understand and explain their capabilities, limitations, benefits, and risks [9, 10]. However, recent developments in mechanistic interpretability have challenged this paradigm, opening the door for a potential cognitive revolution in deep learning. The goal of such work is not (only) to explain specific model *outputs*, but more broadly to interpret the latent representations [11–13] or internal operations and algorithms [14, 15] that are learned in self-supervised pretraining, and to explain how these representations and mechanisms result in observed model behaviors [16–18]. Specifically, we break this work into two broad categories.¹

- *Semantic interpretation*: what latent *representations* are learned and used by models? (See Section 3.)
- *Algorithmic interpretation*: what *operations* and *algorithms* are being implicitly implemented by models (over such representations) to carry out a given behavior? (See Section 4.)

Our goal is to explore the relationship between semantic and algorithmic interpretation, elaborating the different assumptions made by each category of work and how they each approach the notion of interpretation, discussing the parallel challenges faced by each category, and grounding such work in concrete parallels from the history of cognitive science in order to better understand how associated cognitive disciplines have approached the question of understanding, interpreting, and explaining the behavior of intelligent systems.

2 Background

2.1 A Brief History of Deep Learning Interpretation

Since their introduction, neural networks have been considered “black boxes,”² meaning they are not inherently interpretable. The research on deep learning interpretation is concerned with casting light on the “black box” problem in some form or another. In the era of deep learning preceding the development of foundation models (i.e., neural networks pre-trained on large-scale, self-supervised tasks such as LLMs [19]), the terms *interpretation* and *explanation* most often referred to saliency (feature attribution) methods. However, following the paradigm shift toward foundation models, these terms now more often refer to mechanistic interpretation. In this section, we provide a broad overview of both periods and associated paradigms in deep learning interpretation.

Behaviorist Interpretation: The Rise of Post-hoc Explanations The paradigms of behaviorist and mechanistic interpretation of neural networks emerged concurrently, shortly following the modern study of deep learning. For instance, early work in deep learning interpretation discussed explaining the network behavior by identifying the importance of input features for the output, coining the term “saliency maps” [20]; and contemporaneous work studied representations by analyzing different neuron activations within convolutional neural networks [21]. However, the focus on post-hoc behavioral explanations – i.e., explaining why a given model produced a specific outputs [22–24] – gained more traction in the context of the then-dominant paradigm of classification and supervised learning, particularly in the forms of feature attribution methods and counterfactual explanations. Specifically, feature attribution [25–33] analyzes the behavior of the network by identifying input features that are relevant for that output; and counterfactual explanations [34–36] understand the behavior of the network by answering what needs to change in the input for the network to have a particular output. We highlight that

¹In this categorization, we substitute the more common term “interpretability” with “interpretation”, as each category of work is concerned with *interpreting* the inner structure of an otherwise “black box” model, rather than the state of being inherently interpretable (as studied in the area of interpretable machine learning; see Appendix A.1).

²Note that, by “black box” model, we are not simply referring to models that are only accessible via APIs such as ChatGPT; we refer to all models that are not *transparent*, meaning that the operations being performed by the model to transform inputs into outputs cannot be easily interpreted by human practitioners simply by inspecting the model (see Appendix A.1). This includes all but the very simplest neural networks.

these methods are analogous to the behaviorist paradigm in psychology (discussed in Section 1), which studied cognition only in terms of observable behaviors, not internal mental representations or cognitive processing.

Mechanistic Interpretation: The Cognitive Revolution in Interpretability With the rise of generative and self-supervised learning paradigms that have enabled increasingly powerful and task-general foundation models, the focus has shifted toward questions regarding what these models actually learn in pre-training and what implicit algorithms they perform internally, rather than why a model generated a particular output in a downstream task. Such work is often referred to as *mechanistic interpretability*, which is generally defined as the subfield of interpretability research concerned with “reverse engineer[ing] neural networks, similar to how one might reverse engineer a compiled binary computer program” [37]. This description clearly covers implicit algorithms and individual constituent operations, as studied in circuit discovery (see Section 4); but it is less clear precisely how mechanistic interpretability relates to the study of latent representations of human-interpretable concepts, particularly when specific concepts of interest are provided in advance of empirical analysis (as in probing; see Sections 3.2 and 3.3), rather than dynamically discovered from embedding representations (as in dictionary learning; see Section 3.4). In order to avoid conflating these related but fundamentally distinct notions of interpretability and clarify the specific object of analysis for each family of work, we define a taxonomy of interpretation methods according to the study of latent representations (*semantic interpretation*; see Section 3) and operations/algorithms (*algorithmic interpretation*; see Section 4), and discuss the possibility of a unified interpretation framework integrating both perspectives in Section 5.

Complementary Paradigms It is important to recognize that mechanistic interpretation and behavioral explanation are complementary rather than competing. For instance, understanding why a given deep neural network produced a certain output remains crucial, especially in safety-critical domains such as healthcare [38–40]; and monitoring the latent knowledge and capabilities of frontier models has key implications for AI safety [41, 42]. Such directions are deeply intertwined and should continue to inform each other, as they have throughout the history of interpretability. For instance, internal representations have been leveraged for feature attribution explanations in Class Activation Mapping (CAM) [43], where neuron activations are used to explain which input features are relevant to the output; and subnetwork analysis (which is closely related to circuits; see Section 4) has worked to explain behaviors in terms of input features by finding and ablating sparse internal pathways through the network [33, 44]. Most recently, causal probing [16, 17] and dictionary learning [13, 45] have been leveraged for explaining the behavior of LLMs in terms of interpretable latent features (see Sections 3.3 and 3.4, respectively), and circuit discovery has been applied to uncover the interpretable subnetworks of LLMs that correspond to certain behaviors [15, 46] (see Section 4). This interplay mirrors parallel approaches to understanding human intelligence as observed between behavioral and cognitive modes of analysis: just as cognitive neuropsychology can be invaluable in explaining real-world behaviors and pathologies, mechanistic interpretations can inform our understanding of model outputs, and vice versa. Both perspectives are essential for a holistic understanding of neural networks.

2.2 Levels of Analysis

Marr’s levels of analysis [47] have long been a workhorse foundational framework for scientific inquiry in neuroscience and other disciplines of cognitive science [48]. They are as follows:

- The level of **computational theory** provides a mathematical description of the **goal** of an information-processing system in terms of the desired transformation from inputs to outputs.
- The level of **representations and algorithms** is concerned with how the system represents inputs and outputs, and the algorithm it employs to carry out the transformation described by the computational theory.
- The level of **implementation** is concerned with how representations and algorithms are realized in a physical or software medium.

A few prior works have analyzed the role of Marr’s levels of analysis in the context of machine learning, approaching these levels from the perspective of the learning process [49, 50]. For example, Hamrick & Mohamed [49] consider the scenario of Deep Q-Networks [51], formulating the computational level as the task of mapping input observations to output actions that maximize a given reward function, the algorithmic level as the learning algorithm used to train the network,³ and the implementation level as the set of choices for implementing this network in software (e.g., the network architecture, hyperparameters, optimizer, etc.). Useful as such examples may be for interrogating various design decisions in training such a network, they are orthogonal to the question of what is actually being represented by the network or what implicit algorithms are being learned to support its task performance. Indeed, only trivial applications of Marr’s levels to a pre-trained network itself (rather than simply the learning process that produced it) are possible without some notion of interpretation: otherwise, one can only indicate embedding vectors as representations and the forward pass of the network as the algorithm, precluding any analysis of latent properties that are being represented by embeddings and the implicit algorithm that is being carried out in the forward pass.

However, as we will show in the following sections, recent developments in semantic and algorithmic interpretation have enabled the study of these models (and not only the learning process that produced them) at the level of representations and algorithms, where studying representations is a matter of *semantic interpretation*, and studying algorithms is a matter of *algorithmic interpretation* (discussed in Sections 3 and 4, respectively).

3 Semantic Interpretation

We may define *semantic interpretation* as a matter of answering the following question: what latent properties are learned and represented by neural networks, and how do they contribute to observed model behaviors? For example, do vision models learn representations of semantically-related object categories or spatial relations; or do LLMs learn representations of syntactic dependencies or lexical relations?

3.1 Optimization and Search

What inputs maximize the activation of a given neuron? Given that individual neurons are the atomic level of representation for all neural networks, a natural question is, to what extent do single neurons code for distinct properties of interest? An analogous hypothesis in neuroscience is the notion of “gnostic cells”, cells (neurons) which fire only in the context of very specific concepts, such as one’s grandmother (hence the name) [52, 53]. Do we find such neurons in contemporary deep learning systems? The most prominent approach to neuron-level semantic interpretation involves selecting an individual neuron and either searching through a pre-defined query set or generating an input (such as an image) that maximally activates the target neuron, and inspecting these optimized inputs for common features. Early work in this area focused on image recognition networks [21, 54–56], where some approaches operate by searching for patterns that maximize the output of neurons via image optimization [20, 21, 57, 58], and others searched through a pre-defined query set to find inputs that maximized neuron activations [54, 59–61]. For instance, Net2Vec [59] and Network Dissection [60, 61] systematically analyze the relationship between concepts and neurons by analyzing the activations of neurons in response to images in the dataset clustered by human-annotated visual concepts. Later work has investigated how the same paradigm can be applied to interpret individual neurons in language models [62–65] and multimodal vision-language models [66, 67].

3.1.1 Assumptions and Challenges

Levels of Representation Most “optimization and search”-based methods operate at the level of individual neurons, which comes with serious limitations: neural networks encode *distributed* representations where features are encoded by multiple neurons; and even small-scale, simplified “toy” models have also been found to

³For each example provided by Hamrick & Mohamed [49], at the level of representations and algorithms, only algorithms are considered. This is natural outside the context of interpretability, as it is not possible to directly interpret representations from dense neural embeddings, meaning that suitable semantic interpretation methods are a prerequisite for studying representations in addition to algorithms at this level.

exhibit *polysemanticity*, where multiple concepts are represented by a single neuron [68]. As such, the underlying assumption made in interpreting the semantics of only single, isolated neurons – i.e., that there is a one-to-one mapping from neuron activations to values taken by latent properties (analogous to the “gnostic cell” hypothesis) – is in no way guaranteed to hold. Some works have relaxed this assumption by considering combinations of neurons instead of only individual neurons [64, 65], but the broader question remains: is a subset of neurons in a given layer the appropriate level at which to analyze neural representations, or should the activations of all neurons in a given layer (i.e., its embedding space) be taken into account? Another question raised by such work is whether and how different architectures should be taken into account when examining neurons activations – e.g., for Transformer-based models, do self-attention layers merit special consideration [69–71], or should neurons in feed-forward layers also be examined [64, 65, 72, 73]?

Needle in a Haystack Finally, perhaps the greatest concern with analyses centered around the activations of individual neurons (or small sets of such) is that billion-parameter scale models have millions of functional neurons, and this number will continue to scale exponentially alongside the number of parameters in state-of-the-art models. Naturally, it would be computationally intractable to perform detailed analysis with respect to each individual neuron; so how can one determine which neurons are meaningful and merit individual analysis, and which neurons can be ignored? While a number of heuristic approaches have been proposed for targeting neurons of interest (e.g., see [61, 67, 72], *inter alia*), there is currently no generally accepted methodology for determining which neurons are most important for any given analysis, meaning that there is no way to guarantee (or even provide bounds on the probability) that one has correctly targeted the neurons whose activations are most important in studying any given research question.

3.2 Structural Probing

Framework One of the most popular and well-studied approaches to has been *structural probing* [11, 12, 74]. The goal of structural probing is to train auxiliary classifiers (probes) to predict discrete latent properties of inputs from model embeddings. For example, one may train a probe to predict parts-of-speech from LLM token embeddings, so when given each token in the sequence (The, cat, meows, for, dinner), the probe predicts the corresponding parts of speech (determiner, noun, verb, preposition, noun), respectively. A strong form of the underlying assumption here is that a model is “representing” a property if and only if this property can be consistently predicted from embeddings (e.g., if probes can achieve high validation accuracy w.r.t. to the property in question). For example, an early and influential argument in structural probing, the *pipeline hypothesis*, uses probe accuracies over linguistic tasks across BERT [75] layers to argue that BERT processes linguistic properties in the same order as the “classical NLP pipeline”, with surface-level features recognized first, followed by syntactic features, and semantic features recognized last [11, 76–78].

3.2.1 Assumptions and Challenges

Levels of Representation In order to carry out structural probing research, one must first define the class of representations to be probed – or equivalently, the architecture of the probe being trained (e.g., linear probes can only detect properties that are linearly-encoded). The question of which probing architecture should be utilized is a contentious one [12, 79, 80]; and below, we outline several choices of probing architecture as utilized in the literature, supporting arguments in favor of each architecture, and corresponding limitations.

Linear Probes Perhaps the most well-studied probing architecture is the *linear probe* [16, 81–89]. Use of such probes assumes the *linear subspace hypothesis*: that property representations are encoded by linear embedding subspaces [90, 91]. For instance, Concept Activation Vectors (CAV) [82] are vectors in the representation space perpendicular to a linear classification boundary that classifies the representations of a concept versus other input representations. One argument in favor of linear probing is the intuition that neural classifiers must make class-discriminative information linearly separable in their final embedding layer, so probes (particularly over final-layer embeddings) should also be linear [81]. Another motivation is that, given enough training data, sufficiently expressive probes can memorize arbitrary probe tasks irrespective of the model being probed [79], so

the accuracy of a simpler (i.e., less expressive) probe may better reflect the actual content of embeddings rather than the expressiveness of the probe.

Nonlinear Probes An alternative approach to structural probing involves training arbitrarily expressive (nonlinear) probes in order to learn any representation that may be encoded by the model [17, 80, 92, 93]. For instance, [94] extends CAV using kernels to classify concept regions instead of only concept vectors. As neural networks are, by design, highly nonlinear mathematical objects, it is natural to expect that they may encode some properties nonlinearly [93]. This is particularly true of earlier or intermediate layers, where – unlike the final layer – class-discriminative information does not need to be made linearly separable in order to facilitate classification. An additional argument in favor of nonlinear probes is that probes should mirror the architecture of the model being probed, as this more directly reflects the information that is usable by the model [80]. However, as noted above, there have been concerns that highly expressive probes may memorize the mapping from embeddings to properties irrespective of the model being probed [79]; so for more complex probes, there is an increased risk that high probing accuracy is more a reflection of the probe itself than it is of the model being probed. For instance, in an extreme case, consider generative vision-language models that use embeddings from a frozen image encoder and fine-tune an LLM to generate corresponding image captions [95, 96]. In this case, the LLM could be understood as a highly complex and expressive probe in the sense that the LLM is an auxiliary network (cf. probe) stacked on top of a frozen vision encoder (cf. model being probed) and trained to generate text captions (cf. probe task). In this case, it is clear that the generative “probe task” is, in fact, being learned by the probe (LLM) and not the original model (vision encoder), given that the vision encoder is never trained to generate text.

Probing for what? Another important assumption is the choice of properties for which to probe. It is impossible to know *a priori* which properties a model is representing or leveraging in any given context [11, 12, 97]; so for any given probing experiment, it is always possible that one has simply failed to capture whatever properties are most important to the model, leading to potentially misleading results. This is a serious concern for semantic interpretation, given that we cannot reasonably presume to know ahead of time what complete set of properties may be represented and leveraged by models or whether they happen correspond to key properties in human cognition, and there is a long-documented trend toward anthropomorphizing intelligent-seeming models (especially in the context of linguistic systems such as LLMs) [98–100]. For instance, while substantial early work in structural probing studied “classic NLP pipeline” properties [11, 76–78] (as discussed above), there is no particular reason to believe that such properties are the most important features for interpreting the internal representation or explaining the behavior of any given LLM. Additionally, there are many ways to interpret structural probing results [12]: naively, one might simply consult probing accuracies and compare them between properties or across layers; but various works have argued for the necessity of comparing probe predictions against randomized baselines [101], control tasks [79], or to compute information gain using control functions [92].

Correlation does not imply causation. Perhaps the single most important concern with structural probing is that, under this paradigm, it is only possible to measure *what properties can be predicted* from a representation, not *whether (or how) they are actually used* by the model [12, 16, 17, 79]. We discuss a few proposed solutions to this problem in the following sections.

3.3 Causal Probing

Framework One proposed solution to structural probing’s inability to distinguish correlation from causation problem is *causal probing*, where interventions are performed over representations detected by (structural) probes, and the resulting impact on model behaviors is measured in order to study how these properties are used by the model [16, 17, 102]. For instance, *amnesic probing* [16] uses the iterative nullspace projection (INLP) algorithm [84] to remove all information that is linearly-predictive of a given target property from LLM embeddings, then performs this intervention in the model’s forward pass to measure the impact of the removal operation on language modeling performance across a large text corpus to broadly estimate and compare the model’s use of various target properties.

3.3.1 Assumptions and Challenges

Inherited from Structural Probing As in structural probing, causal probing requires one to pre-define the level of representation (neuron-level, linear, or nonlinear) and set of properties for which to probe before any probing experiment can begin. Most intervention methodologies are built to operate at only a single level of representation – typically linear [16, 84, 103, 104], with some work exploring kernelized linear representations [105, 106] – meaning that methods and results from one level cannot be directly adapted or compared to those from other levels. Alternative approaches have been defined using adversarial attacks against arbitrary probing architectures, allowing interventions to target whatever level of representation assumed by the probe [17, 107]; but these approaches come with fewer theoretical guarantees on potential collateral damage to non-targeted properties, as discussed below.

Completeness vs. Selectivity An important observation made by Elazar *et al.* [16] is that properties removed from earlier layers can be *recoverable* by later ones – i.e., when INLP is used to remove information about some target property from the embeddings of a given upstream layer, it is often still possible to train (linear) probes to predict the property from embeddings of a later downstream layer, meaning that these linear interventions are *incomplete*. For such properties, this finding may be taken as evidence against the linear subspace hypothesis discussed above: INLP removes *all* information that is linearly predictive of the target property, so if BERT only encoded these properties linearly, it would not be possible to recover them following an INLP intervention. Later works in causal probing have investigated nonlinear interventions [17, 105, 106], but as in structural probing, there is a tradeoff associated with expressivity: theoretically, the more powerful (and potentially more *complete*) an intervention is, the more “collateral damage” it may also cause to representations more generally [108], in which case the intervention is less *selective* in restricting damage to only the target property.⁴ Thus, for the most general intervention methodologies (such as those proposed by Davies *et al.* [17] and Tucker *et al.* [107], which can be used to manipulate representations detected by any differentiable probe), it is difficult to determine whether any observed changes to model behavior in the presence of interventions is attributable to the model’s representation of the target property or simply to collateral damage (i.e., low selectivity).

Rashomon Effect Finally, as in most studies of causality, causal probing introduces the Rashomon effect [109, 110]: when there are multiple explanations with equal causal efficacy (e.g., if removing information about property A and property B from the model leads to the same impact in its behavior), there is no way to determine which explanation is “correct” (e.g., we cannot say whether the model’s representation of A or B is responsible for its behavior) [111].

3.4 Dictionary Learning

As noted in Sections 3.2 and 3.3, one of the key limitations of both structural and causal probing is that they are fully supervised, meaning that one must pre-define a set of latent properties for which to probe. This means that, even for a perfect (causal) probing methodology, it is always possible that the most important properties leveraged by the model in the course of performing a particular task could be completely missed if one simply does not probe for these particular properties [11, 12, 97]. This is a serious concern for semantic interpretation, given that we cannot reasonably presume to know ahead of time what complete set of properties may be represented and leveraged by a model in any given context.

Dictionary Learning An alternative paradigm in semantic interpretation, *dictionary learning*, removes this presumption by inverting the traditional probing process: instead of directly training a *supervised* probe to predict some latent property of interest from a model’s intermediate embedding representations, the goal of dictionary learning is to train an *unsupervised* probe to decompose embeddings into a sparse combination of features and

⁴Here, we use *selectivity* in the sense described by Elazar *et al.* [16], and not other probing work such as Hewitt & Liang [79], where it instead refers to the gap in performance between probes trained to predict real properties versus randomized “nonsense” properties.

use them to reconstruct the original embeddings, yielding a *dictionary* of features that are useful for sparsely representing embeddings [112–114].

Sparse Auto-Encoders Sparse Auto-Encoders (SAEs) [97, 115] have recently emerged as a powerful and scalable approach to unsupervised probing via dictionary learning [13, 45, 116], performing a nonlinear transformation of input embeddings onto an overcomplete linear basis, allowing them to learn (potentially exponentially many) more features than the dimensionality of the embeddings they are trained on. Where early dictionary learning methods in signal processing were based on sparse coding methods involving Bayesian modeling [112] or convex optimization [113], SAEs carry out dictionary learning using neural networks, improving scalability while maintaining the same goal of learning sparse features for decomposition and reconstruction.

3.4.1 Assumptions and Challenges

Levels of Representation: Superposition As with structural and causal probing above, it is necessary to define the level of representation (e.g., neuron-level, linear, or nonlinear) as the target of analysis. Formally, SAEs (as formulated above) are an unsupervised *nonlinear* probe that project embeddings onto an overcomplete *linear* basis, and thus fall somewhere in-between the linear and nonlinear levels of representation. Instead, SAEs follow the *superposition hypothesis* [68], which argues that models learn to represent more features than they have neurons in a given layer by encoding them via *almost-orthogonal* directions in the embedding space, expanding the number of features that can be represented in the embedding space at the cost of some noise due to interference between non-orthogonal features.⁵ According to this hypothesis, models leverage superposition because the benefits of representing (potentially many) more features with fewer neurons outweighs the cost of filtering out this noise (via nonlinear activation functions) so long as features are sufficiently sparse (leading to less interference on average) [68, 118].

Unsupervised Feature Interpretation While SAEs (and dictionary learning more broadly) remove the need for labeled training data for supervised probes, they effectively transfer the burden of annotation from probe training data to unsupervised feature interpretation, as each feature vector in the dictionary cannot be directly interpreted any more easily than dense embeddings themselves. Rather, each feature must be retroactively interpreted, usually by asking an annotator (either a human or an LLM [119]) to inspect the input samples that maximally activate the feature and annotating it according to whatever feature these samples intuitively appear to have in common [13, 45, 116]. For instance, in the largest SAE study to date (including up to 34 million SAE features learned from embeddings of a frontier LLM), Templeton *et al.* [13] find that one feature learned by an unsupervised probe over a large multilingual, multimodal vision-language model corresponds to the *Golden Gate Bridge*, and that this feature is strongly activated in contexts discussing the bridge in many different languages or for images depicting the bridge, while being weakly activated in contexts discussing (or images depicting) other tourist landmarks in San Francisco or other famous bridges. While such features are indeed interesting, it is not clear what proportion of features have such clear interpretations – how many of these millions of features are as easily interpreted as “Golden Gate Bridge”? At such a large scale, it is not feasible for human annotators to manually interpret features, requiring automated interpretation – e.g., prompting another LLM to explain the relationship between multiple passages that highly activate the same feature, a scalable approach that has been shown to be reasonably well-aligned with human judgments [119]. However, it is important to note that, in contrast to supervised probing methods (where target properties are labeled in advance), unsupervised feature interpretations (whether human- or LLM-annotated) are not *transferable* in that this process must be performed *each time* one analyzes a new model, layer, or trains a new SAE – that is, where probing datasets only require that each input (or parts of inputs, such as individual tokens) be labeled once, dictionary learning requires that one (re-)interpret learned features every time a new dictionary is learned, significantly increasing the burden of annotation and preventing direct comparisons between different models, layers, or dictionary learning methods.

⁵Specifically, with d number of neurons, embeddings can encode $\mathcal{O}(\exp(d))$ features with less than ϵ cosine similarity [117].

Evaluating Interpretations More broadly, even for the most intuitive features, it is not clear precisely how one should evaluate whether any given feature interpretation is correct. At the scale of frontier models and large SAEs, beyond requiring automating feature interpretation, it is also necessary to automate *evaluation* of these interpretations. Thus, even for scalable and high-quality automated techniques, stacking multiple layers of LLM automation for interpretation and evaluation (which may even carried out by the same type of model that is the object of analysis) could lead to compounding errors. Current research has addressed the problems inherent in this paradigm by leveraging *causal interventions* over discovered features [13, 45] (directly analogous to those performed in causal probing over supervised features) in order to observe whether model behavior changes consistently with the hypothesized interpretation of each feature – e.g., when the Golden Gate Bridge feature discussed above is significantly strengthened, the associated LLM responds to many queries with references to the famous bridge where it normally would not. However, as in causal probing, such interventions can only tell us whether a given feature contributes to a particular output, not whether there are other features that contribute just as strongly to the prediction [111], nor the extent to which interventions are *complete* (meaning they fully control all aspects of the target feature) or *selective* (meaning they do not also damage non-targeted features).

4 Algorithmic Interpretation

We may define *algorithmic interpretation* as a matter of answering the following questions: what operations is a given neural network implicitly performing (over representations); what role do these operations play in observed behaviors; and, when these operations are taken in aggregate, what algorithm is being implicitly implemented by the model? Such operations and algorithms are typically understood in terms of *circuits*: “sub-graphs of the network” (weights) that implement a given operation or algorithm, which may be themselves composed to form larger circuits [120]. In principle, circuits can refer to any sub-graph of a neural network; but algorithmic interpretation studying Transformer-based models (the architecture of most modern LLMs and many other foundation models) typically studies circuits at the level of individual attention heads, or compositions of such [14, 46].

4.1 Circuit Discovery

Circuit Discovery The task of finding circuits that faithfully describe a given category of model behaviors is often referred to as *circuit discovery* [15, 46]. The simplest circuits to understand are *end-to-end circuits*, which describe a full path (composed of sub-circuits) from input to output, implementing a complete algorithm. For instance, one of the first circuits identified in a Transformer language model is the *induction circuit* [14, 121]: an end-to-end circuit which, given an input of the form “[A] [B] … [A]” (where [A] and [B] are arbitrary token sequences), determines whether or not to predict that [B] once again follows the second [A] (as it did earlier in the sequence). For instance, given the input “Vernon Dursley and Petunia Durs” (tokenized into [Vern, #on, Durs, #ley, and, Petunia, Durs], where # denotes a token that has been created by splitting an existing word into multiple word pieces), an induction circuit would be tasked with predicting whether or not to follow the second “Durs” token with “#ley”. (See Conmy *et al.* [15] for additional examples of end-to-end circuits that have been discovered in LLMs.)

Causal Interventions As in causal probing and dictionary learning above, a popular strategy in circuit discovery is to explain a circuit’s contribution to a model’s behavior by intervening in its forward pass and study the effect on the model’s predictions. There are two popular strategies for doing so, *knockouts* and *patching*, which we discuss in turn below.

Knockout Knockouts “turn off” a (sub-)circuit by nullifying it in order to determine its contribution to LLM behaviors. Given a circuit, a few approaches to performing knockout have been proposed: *zero ablation* sets the output of the analyzed circuit to zero [121]; whereas *mean ablation* instead sets its output to the mean value of a given reference distribution [46]. The former approach is simpler, as it does not require one to define a reference distribution; but as subsequent circuits may “rely on activation value[s] as an implicit bias term” [46], it is theoretically more sound (and empirically less noisy) to perform knockout by setting to the mean value of a reference distribution where possible.

Patching Patching replaces circuit outputs in the course of computing a *target sequence* with activations from computing a *source sequence*, allowing one to measure whether LLM outputs for the patched target sequence change consistently with the hypothesized function of the circuit in the context of the source sequence [15, 46, 72, 73]. For instance, in the earlier example of an induction circuit given a source sequence of “Vernon Dursley and Petunia Durs”, where the next token predicted by the LLM will be “#ley”, one may patch the circuit outputs from this source sequence into the LLM in its forward pass while processing the target sequence “Thus, the argument must be val” in order to swap its prediction for the next token from “#id” to “#ley”.

4.1.1 Assumptions and Challenges

Circuit Architecture As in semantic interpretation, circuit discovery requires one to target a specific level at which to interpret models – for instance, in probing, this is determined by the expressivity of the selected probe architecture; but in circuit discovery, one must decide on the *atomic* (smallest-scale) sub-graphs considered as possible features or (sub-)circuits. For example, Olah *et al.* [120] starts at the level of individual neuron activations; but given the intractability of considering all neurons in larger models, more recent work targeting LLMs has begun at the level of attention heads [15, 46]. While such tradeoffs are necessary in order to study models at scale, it is important to remember that any operations or algorithms implemented by sub-graphs below or outside the scope of the atomic level – e.g., any analysis of Transformer models built exclusively on attention heads will miss sub-circuits implemented by MLPs [64, 72].

One-to-One Mapping Another important assumption in circuit discovery is that atomic sub-graphs are understood as performing one and only one operation. Given that even the simplest of models are known to often represent multiple properties using the same neuron [68], the assumption that single neurons of LLM architectures can be neatly discretized into individual, atomic operations is suspect; and larger sub-graphs can be difficult to precisely localize and may contain redundant elements [122]. This assumption can be somewhat attenuated with the interventions discussed above, as they can be used to test the extent to which knocking out or patching a given sub-graph leads to the behavior predicted by the hypothesized circuit, evaluating whether or not the sub-graph actually performs the indicated operation. However, this process only solves part of the problem: while it can test whether or not the sub-graph is indeed involved in implementing the observed behavior, it cannot determine whether there are other sub-graphs that may also have a similar effect [111, 123]; and in some cases, knocking out randomized circuits can have a similar effect as knocking out circuits that are precisely calibrated to the target behavior [122].

Pre-Specification Another important limitation of current circuit discovery work is that, to our knowledge, all circuits that have so far been identified in real-world LLMs⁶ have required pre-specifying a full algorithmic description before they can be found. While this significantly reduces computational complexity and thus allows discovery to scale to LLMs [15], it also means that such discovery can generally only move “top-down”: in this case, one cannot discover a circuit that implements an algorithm which has not been explicitly specified ahead of time, and will be less likely to discover intermediate sub-graphs that do not already fit into pre-specified algorithms (a more “bottom-up” approach that has been employed in smaller-scale “toy model” investigations [14, 123, 124]).

Rashomon Effect Finally, perhaps the greatest challenge in circuit discovery is that it is possible to yield multiple distinct circuit descriptions for the same LLM behavior depending on how circuit analysis is carried out [111, 123]. (another instance of the Rashomon effect discussed above). For instance, Zhong *et al.* [123] explore an “algorithmic phase transition” experiment where a form of patching is used to interpolate between source and target circuits, allowing them to characterize how correctly each circuit describes the internal representation as the balance is shifted between the source and target, finding that, in some cases, it appears that a single model may actually be performing the algorithms associated with each circuit simultaneously.

⁶I.e., LLMs that are not small-scale “toy models” trained specifically for the purpose of algorithmic interpretation studies, as in, e.g., Elhage *et al.* [14, 68] and Olsson *et al.* [121].

5 Towards Unifying Semantic and Algorithmic Interpretation

Finally, a few theoretical frameworks have been proposed to simultaneously account for the role of representations and algorithms in model behaviors [17, 18] – i.e., to unify semantic interpretation and algorithmic interpretation under a common framework. Geiger *et al.* [18] proposes a causal abstraction formalism to interpret neural network behaviors in terms of an underlying graphical causal model, where representations are nodes in the causal model and operations over these representations are edges, and measures the faithfulness of the causal model as an explanation of the neural network by computing the instance-level alignment between the network and the causal model under interchange interventions [125]; and Davies *et al.* [17] reformulates this framework at the level of concepts instead of individual instances, extending it to concept-level interventions such as those in causal probing or dictionary learning. Several algorithms have been proposed to empirically deploy causal abstraction frameworks to explain the behavior of large-scale neural networks such as LLMs by learning linear rotational interventions over embedding spaces to perform interchange interventions [126, 127] or substituting interchange interventions for gradient-based attacks against nonlinear causal probes [17]. However, to date, each of these works exhibit serious empirical limitations: they either (1) consider only small-scale, simplified “toy” models and tasks [18, 126]; or (2) intervene only in a single neural network layer [17, 127]; meaning that their ability to describe how real-world models carry out operations that transform representations from layer to layer – i.e., to accomplish algorithmic interpretation – has not yet been empirically demonstrated.

6 Conclusion

In this work, we discussed the parallels between such work and research trends in the history of cognitive science, including Marr’s levels of analysis and the cognitive revolution that birthed the modern cognitive sciences. We explored the relationship between two broad categories of deep learning interpretability research, *semantic interpretation* (what latent properties are represented) and *algorithmic interpretation* (what operations are performed over representations), highlighting the importance of causal analysis across both categories in delivering faithful explanations of model behavior. For each category, we surveyed salient works, analyzed their comparative strengths and weaknesses, clarified underlying assumptions, and outlined key challenges; and we discussed the relationships between these categories of work, focusing on the parallel challenges they each face and how they can compliment each other, concluding by outlining recent efforts to provide a unified framework that can account for both semantic and algorithmic interpretation. Our goal is to facilitate more open, productive conversation of deep learning interpretation by providing a common lexicon of goals, assumptions, and challenges associated with different modes of interpretation, stimulating further research toward more rigorous neural network interpretation, and informing current discourse by consulting lessons from the cognitive sciences.

References

1. Mandler, G. Origins of the cognitive (r) evolution. *Journal of the History of the Behavioral Sciences* **38**, 339–353 (2002).
2. Skinner, B. F. *Verbal behavior* (New York: Appleton-Century-Crofts, 1957).
3. Chomsky, N. A review of BF Skinner's Verbal Behavior. *The Language and Thought Series*, 48–64 (1980).
4. Chomsky, N. *Syntactic structures* (Mouton de Gruyter, 2002).
5. Bruner, J. *A study of thinking* (Routledge, 2017).
6. Miller, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* **63**, 81 (1956).
7. Sperry, R. W. The impact and promise of the cognitive revolution. *American psychologist* **48**, 878 (1993).
8. Miller, G. A. The cognitive revolution: a historical perspective. *Trends in cognitive sciences* **7**, 141–144 (2003).
9. Raji, I. D., Bender, E. M., Paullada, A., Denton, E. & Hanna, A. AI and the everything in the whole wide world benchmark. *arXiv preprint arXiv:2111.15366*. <https://doi.org/10.48550/arXiv.2111.15366> (2021).
10. Mahowald, K. *et al.* Dissociating language and thought in large language models: a cognitive perspective. *arXiv preprint arXiv:2301.06627*. <https://doi.org/10.48550/arXiv.2301.06627> (2023).
11. Rogers, A., Kovaleva, O. & Rumshisky, A. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics* **8**, 842–866. <https://aclanthology.org/2020.tacl-1.54> (2020).
12. Belinkov, Y. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics* **48**, 207–219. <https://aclanthology.org/2022.cl-1.7> (Mar. 2022).
13. Templeton, A. *et al.* Scaling Monosematicity: Extracting Interpretable Features from Claude 3 Sonnet. *Transformer Circuits Thread*. <https://transformer-circuits.pub/2024/scaling-monosematicity/index.html> (2024).
14. Elhage, N. *et al.* A mathematical framework for transformer circuits. *Transformer Circuits Thread* **1** (2021).
15. Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S. & Garriga-Alonso, A. Towards Automated Circuit Discovery for Mechanistic Interpretability in Advances in Neural Information Processing Systems (eds Oh, A. *et al.*) **36** (Curran Associates, Inc., 2023), 16318–16352. https://proceedings.neurips.cc/paper_files/paper/2023/file/34e1dbe95d34d7ebaef99b9bcaeb5b2be-Paper.pdf
16. Elazar, Y., Ravfogel, S., Jacovi, A. & Goldberg, Y. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *Transactions of the Association for Computational Linguistics* **9**, 160–175. <https://aclanthology.org/2021.tacl-1.10> (2021).
17. Davies, A., Jiang, J. & Zhai, C. Competence-Based Analysis of Language Models. *arXiv preprint arXiv:2303.00333* (2023).
18. Geiger, A., Potts, C. & Icard, T. Causal Abstraction for Faithful Model Interpretation. *arXiv preprint arXiv:2301.04709* (2023).
19. Bommasani, R. *et al.* On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).
20. Simonyan, K., Vedaldi, A. & Zisserman, A. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps* in Workshop at International Conference on Learning Representations (2014).
21. Zeiler, M. D. & Fergus, R. *Visualizing and understanding convolutional networks* in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I* **13** (2014), 818–833.

-
22. Lipton, Z. C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* **16**, 31–57 (2018).
23. Doran, D., Schulz, S. & Besold, T. R. What does explainable AI really mean? A new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794* (2017).
24. Arrieta, A. B. *et al.* Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion* **58**, 82–115 (2020).
25. Lundberg, S. M. & Lee, S. I. *A unified approach to interpreting model predictions in Advances in Neural Information Processing Systems* (2017). arXiv: 1705.07874.
26. Sundararajan, M., Taly, A. & Yan, Q. *Axiomatic attribution for deep networks* in *34th International Conference on Machine Learning, ICML 2017* (2017). arXiv: 1703.01365.
27. Sundararajan, M. & Najmi, A. The many Shapley values for model explanation. *37th International Conference on Machine Learning, ICML 2020* (2020).
28. Ribeiro, M. T., Singh, S. & Guestrin, C. "Why should i trust you?" *Explaining the predictions of any classifier* in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 13-17-August-2016* (Association for Computing Machinery, New York, New York, USA, Aug. 2016), 1135–1144. ISBN: 9781450342322. arXiv: 1602 . 04938. <http://dl.acm.org/citation.cfm?doid=2939672.2939778>.
29. Fong, R., Patrick, M. & Vedaldi, A. *Understanding deep networks via extremal perturbations and smooth masks* in *Proceedings of the IEEE International Conference on Computer Vision* (2019), 2950–2958.
30. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).
31. Selvaraju, R. R. *et al.* *Grad-cam: Visual explanations from deep networks via gradient-based localization* in *Proceedings of the IEEE international conference on computer vision* (2017), 618–626.
32. Khakzar, A., Khorsandi, P., Nobahari, R. & Navab, N. *Do explanations explain? model knows best* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 10244–10253.
33. Bach, S. *et al.* On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* **10**, e0130140 (2015).
34. Wachter, S., Mittelstadt, B. & Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* **31**, 841 (2017).
35. Lang, O. *et al.* *Explaining in style: Training a gan to explain a classifier in stylespace* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 693–702.
36. Ribeiro, M. T., Wu, T., Guestrin, C. & Singh, S. Beyond accuracy: Behavioral testing of NLP models with CheckList. *arXiv preprint arXiv:2005.04118* (2020).
37. Olah, C. *Mechanistic Interpretability, Variables, and the Importance of Interpretable Bases* 2022. <https://transformer-circuits.pub/2022/mech-interp-essay/index.html>.
38. Reddy, S. Explainability and artificial intelligence in medicine. *The Lancet Digital Health* **4**, e214–e215 (2022).
39. Petch, J., Di, S. & Nelson, W. Opening the black box: the promise and limitations of explainable machine learning in cardiology. *Canadian Journal of Cardiology* **38**, 204–213 (2022).
40. Thirunavukarasu, A. J. *et al.* Large language models in medicine. *Nature medicine* **29**, 1930–1940 (2023).
41. Hendrycks, D., Carlini, N., Schulman, J. & Steinhardt, J. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916* (2021).
42. Zou, A. *et al.* Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405* (2023).
43. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. & Torralba, A. *Learning deep features for discriminative localization* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 2921–2929.

-
44. Khakzar, A. *et al.* *Neural Response Interpretation through the Lens of Critical Pathways* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 13528–13538.
45. Bricken, T. *et al.* Towards Monosematicity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*. <https://transformer-circuits.pub/2023/monosemantic-features/index.html> (2023).
46. Wang, K. R., Variengien, A., Conmy, A., Shlegeris, B. & Steinhardt, J. *Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small* in *The Eleventh International Conference on Learning Representations* (2023). <https://openreview.net/forum?id=NpsVSN6o4ul>.
47. Marr, D. *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information* (W.H. Freeman, 1982).
48. Niv, Y. & Langdon, A. Reinforcement learning with Marr. *Current opinion in behavioral sciences* **11**, 67–73 (2016).
49. Hamrick, J. & Mohamed, S. *Levels of analysis for machine learning* in (2020).
50. Sawant, S. P. & Singh, S. Understanding attention: in minds and machines. *arXiv preprint arXiv:2012.02659* (2020).
51. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *nature* **518**, 529–533 (2015).
52. Gross, C. G. Genealogy of the “grandmother cell”. *The Neuroscientist* **8**, 512–518 (2002).
53. Quiroga, R. Gnostic cells in the 21st century. *Acta Neurobiologiae Experimentalis* **73**, 463–471 (2013).
54. Erhan, D., Bengio, Y., Courville, A. & Vincent, P. Visualizing higher-layer features of a deep network. *University of Montreal* **1341**, 1 (2009).
55. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).
56. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. & Torralba, A. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856* (2014).
57. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T. & Clune, J. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems* **29** (2016).
58. Olah, C., Mordvintsev, A. & Schubert, L. Feature visualization. *Distill* **2**, e7 (2017).
59. Fong, R. & Vedaldi, A. *Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 8730–8738.
60. Bau, D., Zhou, B., Khosla, A., Oliva, A. & Torralba, A. *Network dissection: Quantifying interpretability of deep visual representations* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 6541–6549.
61. Bau, D. *et al.* *Visualizing and Understanding Generative Adversarial Networks* in *International Conference on Learning Representations* (2019). https://openreview.net/forum?id=Hyg_X2C5FX.
62. Karpathy, A., Johnson, J. & Fei-Fei, L. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078* (2015).
63. Dalvi, F. *et al.* *What is one grain of sand in the desert? analyzing individual neurons in deep nlp models* in *Proceedings of the AAAI Conference on Artificial Intelligence* **33** (2019), 6309–6317.
64. Geva, M., Schuster, R., Berant, J. & Levy, O. *Transformer Feed-Forward Layers Are Key-Value Memories* in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (eds Moens, M.-F., Huang, X., Specia, L. & Yih, S. W.-t.) (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, Nov. 2021), 5484–5495. <https://aclanthology.org/2021.emnlp-main.446>.
65. Dai, D. *et al.* *Knowledge Neurons in Pretrained Transformers* in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (eds Muresan, S., Nakov, P. & Villavicencio, A.) (Association for Computational Linguistics, Dublin, Ireland, May 2022), 8493–8502. <https://aclanthology.org/2022.acl-long.581>.

-
66. Hernandez, E. *et al.* *Natural language descriptions of deep visual features* in *International Conference on Learning Representations* (2021).
67. Schwettmann, S., Chowdhury, N., Klein, S., Bau, D. & Torralba, A. *Multimodal neurons in pretrained text-only transformers* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), 2862–2867.
68. Elhage, N. *et al.* Toy models of superposition. *arXiv preprint arXiv:2209.10652* (2022).
69. Voita, E., Talbot, D., Moiseev, F., Sennrich, R. & Titov, I. *Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned* in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (eds Korhonen, A., Traum, D. & Màrquez, L.) (Association for Computational Linguistics, Florence, Italy, July 2019), 5797–5808. <https://aclanthology.org/P19-1580>.
70. Vig, J. & Belinkov, Y. *Analyzing the Structure of Attention in a Transformer Language Model* in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (eds Linzen, T., Chrupała, G., Belinkov, Y. & Hupkes, D.) (Association for Computational Linguistics, Florence, Italy, Aug. 2019), 63–76. <https://aclanthology.org/W19-4808>.
71. Clark, K., Khandelwal, U., Levy, O. & Manning, C. D. *What Does BERT Look at? An Analysis of BERT’s Attention* in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (eds Linzen, T., Chrupała, G., Belinkov, Y. & Hupkes, D.) (Association for Computational Linguistics, Florence, Italy, Aug. 2019), 276–286. <https://aclanthology.org/W19-4828>.
72. Meng, K., Bau, D., Andonian, A. & Belinkov, Y. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems* **35**, 17359–17372 (2022).
73. Meng, K., Sharma, A. S., Andonian, A. J., Belinkov, Y. & Bau, D. *Mass-Editing Memory in a Transformer* in *The Eleventh International Conference on Learning Representations* (2023). <https://openreview.net/forum?id=MkbcAHIYgyS>.
74. Belinkov, Y., Gehrman, S. & Pavlick, E. *Interpretability and Analysis in Neural NLP* in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts* (Association for Computational Linguistics, Online, July 2020), 1–5. <https://aclanthology.org/2020.acl-tutorials.1>.
75. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Association for Computational Linguistics, Minneapolis, Minnesota, June 2019), 4171–4186. <https://aclanthology.org/N19-1423>.
76. Tenney, I., Das, D. & Pavlick, E. *BERT Redisovers the Classical NLP Pipeline* in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Florence, Italy, July 2019), 4593–4601. <https://aclanthology.org/P19-1452>.
77. Jawahar, G., Sagot, B. & Seddah, D. *What does BERT learn about the structure of language?* in *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics* (2019).
78. Niu, J., Lu, W. & Penn, G. *Does BERT Rediscover a Classical NLP Pipeline?* in *Proceedings of the 29th International Conference on Computational Linguistics* (2022), 3143–3153.
79. Hewitt, J. & Liang, P. *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, Nov. 2019), 2733–2743. <https://aclanthology.org/D19-1275>.
80. Pimentel, T., Valvoda, J., Stoehr, N. & Cotterell, R. The Architectural Bottleneck Principle. *arXiv preprint arXiv:2211.06420*. <https://doi.org/10.48550/arXiv.2211.06420> (2022).
81. Alain, G. & Bengio, Y. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644* (2016).

-
82. Kim, B. *et al.* *Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)* in *International conference on machine learning* (2018), 2668–2677.
83. Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E. & Smith, N. A. *Linguistic Knowledge and Transferability of Contextual Representations* in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (eds Burstein, J., Doran, C. & Solorio, T.) (Association for Computational Linguistics, Minneapolis, Minnesota, June 2019), 1073–1094. <https://aclanthology.org/N19-1112>.
84. Ravfogel, S., Elazar, Y., Gonen, H., Twiton, M. & Goldberg, Y. *Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection* in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Online, July 2020), 7237–7256. <https://aclanthology.org/2020.acl-main.647>.
85. Schwettmann, S. *et al.* *Toward a visual concept vocabulary for gan latent space* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 6804–6812.
86. Park, K., Choe, Y. J. & Veitch, V. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658* (2023).
87. Marks, S. & Tegmark, M. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824* (2023).
88. Tigges, C., Hollinsworth, O. J., Geiger, A. & Nanda, N. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154* (2023).
89. Nanda, N., Lee, A. & Wattenberg, M. *Emergent Linear Representations in World Models of Self-Supervised Sequence Models* in *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP* (eds Belinkov, Y. *et al.*) (Association for Computational Linguistics, Singapore, Dec. 2023), 16–30. <https://aclanthology.org/2023.blackboxnlp-1.2>.
90. Vargas, F. & Cotterell, R. *Exploring the Linear Subspace Hypothesis in Gender Bias Mitigation* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Online, Nov. 2020), 2902–2913. <https://aclanthology.org/2020.emnlp-main.232>.
91. Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V. & Kalai, A. T. *Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings* in *Advances in Neural Information Processing Systems* (eds Lee, D., Sugiyama, M., Luxburg, U., Guyon, I. & Garnett, R.) **29** (Curran Associates, Inc., 2016). https://proceedings.neurips.cc/paper_files/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf
92. Pimentel, T. *et al.* *Information-Theoretic Probing for Linguistic Structure* in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Online, July 2020), 4609–4622. <https://aclanthology.org/2020.acl-main.420>.
93. White, J. C., Pimentel, T., Saphra, N. & Cotterell, R. *A Non-Linear Structural Probe* in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (eds Toutanova, K. *et al.*) (Association for Computational Linguistics, Online, June 2021), 132–138. <https://aclanthology.org/2021.naacl-main.12>.
94. Crabbé, J. & van der Schaar, M. Concept activation regions: A generalized framework for concept-based explanations. *Advances in Neural Information Processing Systems* **35**, 2590–2607 (2022).
95. Zhang, P. *et al.* *VinVL: Revisiting Visual Representations in Vision-Language Models* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), 5579–5588.
96. Zhai, X. *et al.* *LiT: Zero-Shot Transfer With Locked-Image Text Tuning* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), 18123–18133.
97. Yun, Z., Chen, Y., Olshausen, B. & LeCun, Y. *Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors* in *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures* (eds Agirre, E., Apidianaki, M. & Vulic, I.) (Association for Computational Linguistics, Online, June 2021), 1–10. <https://aclanthology.org/2021.deelio-1.1>.

-
98. Weizenbaum, J. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM* **9**, 36–45 (1966).
99. Turkle, S. Authenticity in the age of digital companions. *Interaction studies* **8**, 501–517 (2007).
100. Switzky, L. ELIZA effects: Pygmalion and the early development of artificial intelligence. *Shaw* **40**, 50–68 (2020).
101. Tenney, I. *et al.* What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316* (2019).
102. Lasri, K., Pimentel, T., Lenci, A., Poibeau, T. & Cotterell, R. *Probing for the Usage of Grammatical Number in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, Dublin, Ireland, May 2022), 8818–8831. <https://aclanthology.org/2022.acl-long.603>.
103. Ravfogel, S., Prasad, G., Linzen, T. & Goldberg, Y. *Counterfactual Interventions Reveal the Causal Effect of Relative Clause Representations on Agreement Prediction in Proceedings of the 25th Conference on Computational Natural Language Learning* (eds Bisazza, A. & Abend, O.) (Association for Computational Linguistics, Online, Nov. 2021), 194–209. <https://aclanthology.org/2021.conll-1.15>.
104. Ravfogel, S., Twiton, M., Goldberg, Y. & Cotterell, R. D. *Linear adversarial concept erasure* in *International Conference on Machine Learning* (2022), 18400–18421.
105. Ravfogel, S., Vargas, F., Goldberg, Y. & Cotterell, R. *Adversarial Concept Erasure in Kernel Space* in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, Dec. 2022), 6034–6055. <https://aclanthology.org/2022.emnlp-main.405>.
106. Shao, S., Ziser, Y. & Cohen, S. B. *Gold Doesn't Always Glitter: Spectral Removal of Linear and Nonlinear Guarded Attribute Information* in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics* (eds Vlachos, A. & Augenstein, I.) (Association for Computational Linguistics, Dubrovnik, Croatia, May 2023), 1611–1622. <https://aclanthology.org/2023.eacl-main.118>.
107. Tucker, M., Qian, P. & Levy, R. *What if This Modified That? Syntactic Interventions with Counterfactual Embeddings* in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (Association for Computational Linguistics, Online, Aug. 2021), 862–875. <https://aclanthology.org/2021.findings-acl.76>.
108. Zhao, H. *et al.* Fundamental limits and tradeoffs in invariant representation learning. *The Journal of Machine Learning Research* **23**, 15356–15404 (2022).
109. Breiman, L. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science* **16**, 199–231 (2001).
110. Hancov-Li, L. *Robustness in machine learning explanations: Does it matter?* in *Proceedings of the 2020 conference on fairness, accountability, and transparency* (2020), 640–647.
111. Mueller, A. *Missed Causes and Ambiguous Effects: Counterfactuals Pose Challenges for Interpreting Neural Networks* in *ICML 2024 Workshop on Mechanistic Interpretability* (2024). <https://openreview.net/forum?id=pJs3ZiKBM5>.
112. Lewicki, M. S. & Sejnowski, T. J. Learning Overcomplete Representations. *Neural Computation* **12**, 337–365. ISSN: 0899-7667. eprint: <https://direct.mit.edu/neco/article-pdf/12/2/337/814391/089976600300015826.pdf>. <https://doi.org/10.1162/089976600300015826> (Feb. 2000).
113. Lee, H., Battle, A., Raina, R. & Ng, A. *Efficient sparse coding algorithms* in *Advances in Neural Information Processing Systems* (eds Schölkopf, B., Platt, J. & Hoffman, T.) **19** (MIT Press, 2006). https://proceedings.neurips.cc/paper_files/paper/2006/file/2d71b2ae158c7c5912cc0bbde2bb9d95-Paper.pdf

-
114. Faruqui, M., Tsvetkov, Y., Yogatama, D., Dyer, C. & Smith, N. A. *Sparse Overcomplete Word Vector Representations* in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (eds Zong, C. & Strube, M.) (Association for Computational Linguistics, Beijing, China, July 2015), 1491–1500. <https://aclanthology.org/P15-1144>.
115. Subramanian, A., Pruthi, D., Jhamtani, H., Berg-Kirkpatrick, T. & Hovy, E. *Spine: Sparse interpretable neural embeddings* in *Proceedings of the AAAI conference on artificial intelligence* **32** (2018).
116. Cunningham, H., Ewart, A., Riggs, L., Huben, R. & Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600* (2023).
117. Dasgupta, S. & Gupta, A. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms* **22**, 60–65 (2003).
118. Scherlis, A., Sachan, K., Jermyn, A. S., Benton, J. & Shlegeris, B. Polysemanticity and capacity in neural networks. *arXiv preprint arXiv:2210.01892* (2022).
119. Bills, S. *et al.* *Language models can explain neurons in language models* <https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html>. 2023.
120. Olah, C. *et al.* Zoom in: An introduction to circuits. *Distill* **5**, e00024–001 (2020).
121. Olsson, C. *et al.* In-context learning and induction heads. *arXiv preprint arXiv:2209.11895* (2022).
122. Shi, C. *et al.* *Hypothesis Testing the Circuit Hypothesis in LLMs* in *ICML 2024 Workshop on Mechanistic Interpretability* (2024).
123. Zhong, Z., Liu, Z., Tegmark, M. & Andreas, J. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *arXiv preprint arXiv:2306.17844* (2023).
124. Nanda, N., Chan, L., Lieberum, T., Smith, J. & Steinhardt, J. *Progress measures for grokking via mechanistic interpretability* in *The Eleventh International Conference on Learning Representations* (2023). <https://openreview.net/forum?id=9XFSbDPmdW>.
125. Geiger, A. *et al.* *Inducing Causal Structure for Interpretable Neural Networks* in *Proceedings of the 39th International Conference on Machine Learning* (eds Chaudhuri, K. *et al.*) **162** (PMLR, 17–23 Jul 2022), 7324–7338. <https://proceedings.mlr.press/v162/geiger22a.html>.
126. Geiger, A., Wu, Z., Potts, C., Icard, T. & Goodman, N. *Finding alignments between interpretable causal variables and distributed neural representations* in *Causal Learning and Reasoning* (2024), 160–187.
127. Wu, Z., Geiger, A., Icard, T., Potts, C. & Goodman, N. Interpretability at scale: Identifying causal mechanisms in alpaca. *Advances in Neural Information Processing Systems* **36** (2024).
128. Doshi-Velez, F. & Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).
129. Broniatowski, D. A. *et al.* Psychological foundations of explainability and interpretability in artificial intelligence. *NIST, Tech. Rep* (2021).
130. Adadi, A. & Berrada, M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access* **6**, 52138–52160 (2018).

A Supplementary Background

A.1 Interpretable Machine Learning

Many definitions of “interpretability” have been proposed to describe methods that fall outside our analysis in this work [22, 24, 128, 129] concerning either (1) interpretability as an inherent feature of human-understandable methods, or (2) input-output explanations (e.g., to what input features is a given output attributable) of *supervised* deep learning models. In contrast, in this work we are concerned with the study of internal mechanisms and latent representations learned by self-supervised (foundation) models that has come to characterize much of the inter-

pretability landscape, and is now most often referred to under the broad umbrella of *mechanistic interpretability* (see Section 2.1).

For example, the most widely-cited definition of interpretability is provided by Lipton [22], who provides two general categories of interpretability. The first is *transparency*, an inherent quality of models that can be fully decomposed into clear, comprehensible operations (e.g., decision trees or rule-based expert systems). The second is *post-hoc explanations*, a family of techniques to explain specific outputs of an otherwise opaque (“black box”) model (e.g., saliency maps [20]). Naturally, any interpretability work studying neural networks must fall into the category of post-hoc explanations, as neural networks are inherently opaque. However, the notion of post-hoc explanation is, on its own, insufficient to describe the wealth of interpretability research that has developed around foundation models: the categories of post-hoc explanation techniques outlined by Lipton [22] predate the era of self-supervised foundation models that have come to dominate many areas of study in AI and ML, and do not apply to contemporary internal analysis techniques such as probing (see Section 3.2) or circuit discovery (see Section 4), where the goal is not necessarily to explain specific model outputs, but rather to interpret the internal representations, operations, and algorithms that foundation models learn in self-supervised pretraining. While several works have aimed to update or refine Lipton [22]’s interpretability taxonomy and definitions in various ways (e.g., see Doran *et al.* [23], Arrieta *et al.* [24], and Adadi & Berrada [130]), none of these have addressed the disconnect between *post-hoc explanations* as a matter of explaining specific outputs, and the abundance of recent work oriented around the broader, more internal notion of interpretability that describes the study of latent representations and operations learned by foundation models.