A Practical Review of Mechanistic Interpretability for Transformer-Based Language Models

Daking Rai[1], Yilun Zhou[2], Shi Feng[3,4], Abulhair Saparov[3,5], Ziyu Yao[1]
[1]George Mason University, [2]Salesforce Research,
[3]New York University, [4]George Washington University, [5]Purdue University
{drai2, ziyuyao}@gmu.edu, yilun.zhou@salesforce.com, as17582@nyu.edu, shi.feng@gwu.edu
https://github.com/Dakingrai/awesome-mechanistic-interpretability-lm-papers

Abstract

Mechanistic interpretability (MI) is an emerging sub-field of interpretability that seeks to understand a neural network model by reverse-engineering its internal computations. Recently, MI has garnered significant attention for interpreting transformer-based language models (LMs), resulting in many novel insights yet introducing new challenges. However, there has not been work that comprehensively reviews these insights and challenges, particularly as a guide for newcomers to this field. To fill this gap, we present a comprehensive survey outlining fundamental objects of study in MI, techniques that have been used for its investigation, approaches for evaluating MI results, and significant findings and applications stemming from the use of MI to understand LMs. In particular, we present a roadmap for beginners to navigate the field and leverage MI for their benefit. Finally, we also identify current gaps in the field and discuss potential future directions.

1 Introduction

In recent years, transformer-based language models (LMs) have achieved remarkable success in a wide range of tasks (Radford et al., 2019; Brown et al., 2020; Achiam et al., 2023; Touvron et al., 2023; Bubeck et al., 2023). Alongside these advancements, there are growing concerns over the safety, alignment, reliability, and robustness of their usage and development (Chang et al., 2024; Yao et al., 2024; Weidinger et al., 2022), especially as they are increasingly implemented in real-world applications. These concerns primarily stem from our limited understanding of these LMs and the difficulty in interpreting their behavior.

Recently, mechanistic interpretability (MI) has emerged as a promising technique that fills the gap in the research field of interpretability. This is a line of methods that interpret a model by reverse-engineering the underlying computation into human-understandable mechanisms (Olah et al., 2020; Elhage et al., 2021). It has shown promise in providing insights into the functions of LM components (e.g., neurons, attention heads), offering mechanistic explanations for various LM behaviors, and enabling users to leverage the explanations to enhance an LM's utilization (Wang et al., 2022a; Marks et al., 2024; Templeton et al., 2024). Despite the promise, however, there are concerns on the scalability and generalizability of MI and its helpfulness in addressing critical problems in AI safety (Räuker et al., 2023; Casper, 2023).

Observing these promises and challenges, we aim to provide a comprehensive review of MI in its applications to interpret transformer-based LMs. A *taxonomy* is present in Figure 1. There exist surveys on relevant topics, but they differ from ours in a number of key aspects. For example, while Räuker et al. (2023) discussed interpretability approaches broadly for all types of deep neural networks, we target an in-depth review of MI for transformer-based LMs. Similarly, Zhao et al. (2024) presented a broad survey of LM interpretability but did not focus on MI. Concurrent with our work, Ferrando et al. (2024) reviewed techniques developed for understanding the inner workings of transformer-based LMs but did not particularly focus on MI, and Bereska and Gavves (2024) emphasized MI advances for AI safety.

Our paper is organized as follows. We first present the background knowledge needed to understand this survey (Section 2). Then, we review the key developments in MI for LMs, including the fundamental objects of study, techniques, and evaluation (Sections 3-5). We then particularly present a *beginner's roadmap* in Section 6, which summarizes actionable items for newcomers to use MI for their benefits. This is then followed by an overview of findings and applications of MI in Section 7. Finally, we conclude the survey by discussing challenges and future work in this field.

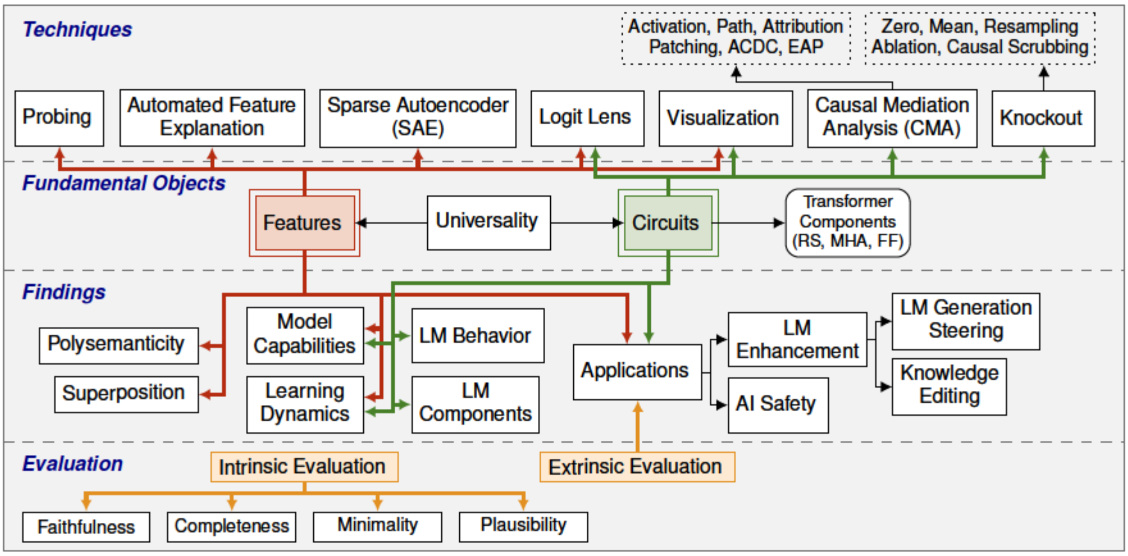


Figure 1: **Taxonomy** of the survey. Blocks are clickable to be directed to the corresponding sections.

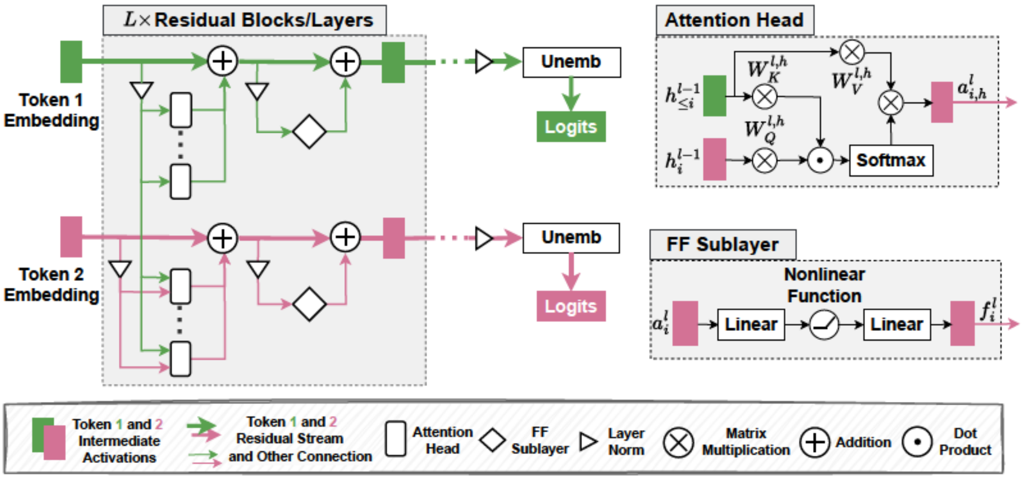


Figure 2: Architecture of transformer-based LMs.

2 Background: Transformer-based LM

A transformer-based LM (Vaswani et al., 2017) M takes input tokens $X = (x_1, ..., x_n)$ and outputs a vector in $\mathbb{R}^{|\mathcal{V}|}$, a probability distribution over the vocabulary $\mathcal{V}$, to predict the next token x_{n+1}. The model refines the representation of each token x_i layer by layer (Figure 2). In the first layer, h_i^0 is an embedding vector of x_i, resulting from a lookup operation in an embedding matrix $W_E \in \mathbb{R}^{|\mathcal{V}|\times d}$.[1] This representation is then updated layer-by-layer through the calculations of *multi-head attention (MHA)* and *feed-forward (FF)* sublayers in each layer, i.e.,

$$h_i^l = h_i^{l-1} + a_i^l + f_i^l, \tag{1}$$

where h_i^l denotes the representation of token x_i at layer l, a_i^l is the attention output from the MHA sublayer, and f_i^l is the output from the FF sublayer. The sequence of h_i^l across the layers is also referred to as the *residual stream* (RS) of transformer in literature (Elhage et al., 2021).

Briefly, the MHA sublayer with H attention heads is implemented via

$$a_i^l = \mathtt{concat}(a_{i,0}^l, .., a_{i,H}^l)W_O^l, \tag{2}$$

$$a_{i,h}^l = \mathtt{softmax}\left(\frac{(h_i^{l-1}W_Q^{l,h})(h_{\leq i}^{l-1}W_K^{l,h})^\top}{\sqrt{d_k}}\right) \cdot (h_{\leq i}^{l-1}W_V^{l,h}), \tag{3}$$

[1]For brevity, we omit components such as position embedding and layer normalization in transformer, as they will not affect our discussion of MI. Readers should refer to Vaswani et al. (2017) for a complete description.

where $a_{i,h}^l$ is the attention output from the h-th head, $W_Q^{l,h}, W_K^{l,h}, W_V^{l,h}$ are the query, key, and value (learned) projection matrices, respectively, and W_O^l projects the concatenated attention outputs from all heads to the model dimension d.

The FF sublayer then performs two linear transformations over a_i^l with an element-wise non-linear function σ between them, i.e.,

$$f_i^l = W_v^l \sigma(W_k^l a_i^l + b_k^l) + b_v^l, \tag{4}$$

where W_v^l, W_k^l, b_k^l, and b_v^l are learned parameter matrices and biases. Finally, the RS of x_n at the final layer, h_n^L, is projected into a probability distribution over $\mathcal{V}$ by applying an *unembedding* matrix $W_U \in \mathbb{R}^{d \times |\mathcal{V}|}$ and a softmax operation.

3 Fundamental Objects of Study

MI is a bottom-up approach that interprets LMs by decomposing them into smaller components and more elementary computations. Following Olah et al. (2020), one of the earliest studies in MI, we categorize research of MI into three areas: the study of features, circuits, and their universality.

3.1 Features

A *feature* is a human-interpretable input property that is encoded in a model's activation.[2] For instance, a neuron or a set of neurons that consistently activates for French text can be interpreted as a "French text detector" feature (Gurnee et al., 2023).[3] MI aims to interpret LMs in terms of the independent features or components representing these features. To this end, Elhage et al. (2022b) proposed the *linear representation hypothesis*, which posits that neural networks have two properties – *linearity*, i.e. the network's activation space consists of meaningful (linear) vectors, each representing a feature, and *decomposability*, i.e., network activations can be decomposed and described in terms of these independent features. This hypothesis resonates with earlier research that also shows linearity in word embeddings (e.g. $\overrightarrow{man} - \overrightarrow{woman} \approx \overrightarrow{king} - \overrightarrow{queen}$) (Mikolov et al., 2013). In Section 7.1, we will review some identified features in LMs and how they are represented.

[2]We use "activations" and "representations" interchangeably to refer to the intermediate outputs of LM components.

[3]We refer to each element in the activation as a neuron, which implies that a neuron is a basis-aligned direction in the activation space (Marks et al., 2024).

3.2 Circuits

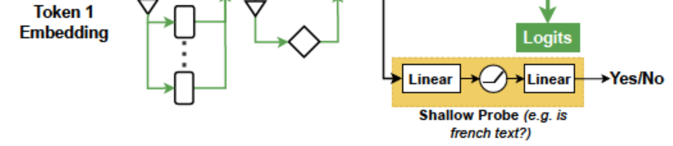
While the study of features helps us to understand what information is encoded in a model's activations, it does not inform us of how these features are extracted from the input and then processed to implement specific LM behaviors (e.g., reasoning). MI fills this gap by studying "circuits", which are meaningful computational pathways that connect features.

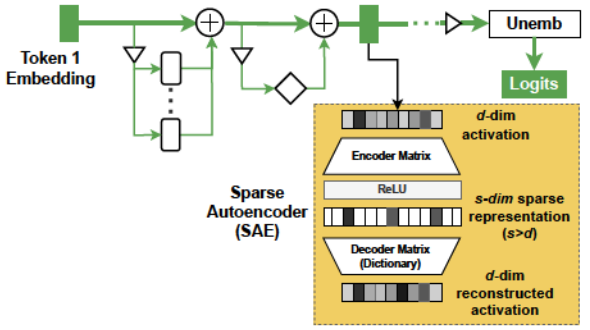
More formally, if we view an LM M as a computational graph with features as nodes and the weighted connections between them as edges, a circuit is a sub-graph of M responsible for implementing specific LM behaviors (Wang et al., 2022a). Additionally, although circuits were initially defined as connections between features (Olah et al., 2020), subsequent studies have generalized them as connections between the activation outputs of transformer components (Olsson et al., 2022; Wang et al., 2022a). Therefore, we include research on interpreting transformer components, both individually (e.g., interpreting RS in Eq 1 connecting the activations of MHA, FF, and the RS from earlier layers) and across multiple components (e.g., the induction circuit in Figure 3), as circuit study.

An example circuit discovered by Elhage et al. (2021) in a toy LM is shown in Figure 3. This is an induction circuit consisting of outputs of two attention heads (previous token head and induction head) as nodes and connection between them as edges of the circuit. The circuit implements the task of detecting and continuing repeated subsequences in the input (e.g., *Mr D urs ley was thin and bold. Mr D -> urs*), where the **previous token head** encodes the information *"urs" follows the "D" token* in the RS, which is then read by the **induction head** to promote *"urs"* as the next token prediction.

3.3 Universality

For any feature or circuit that we have identified in an LM in one task, the critical question arises: Do they similarly exist in other LMs or in other tasks? The investigation into this question has then given rise to the notion of *universality*, i.e., the extent to which similar features and circuits are formed across different LMs and tasks (Olah et al., 2020; Gurnee et al., 2024). The implications of universal features and circuits can be significant. For instance, many studies (Olsson et al., 2022; Elhage et al., 2022a,b) on features and circuits were





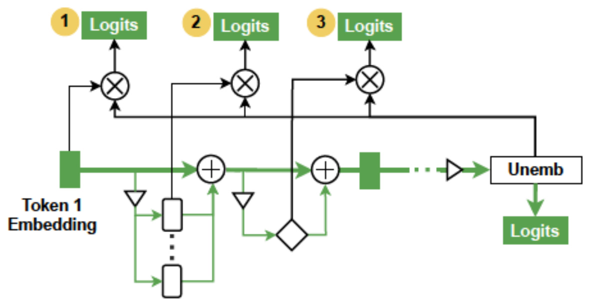


Figure 4: Logit lens implementation at (1) RS, (2) attention head, and (3) FF sublayer.

performed with only toy or small LMs. If these features and circuits are universal, the insights from these studies can be transferred to other unexamined LMs and potentially state-of-the-art large LMs (LLMs). However, if they are not universal, then a significant amount of independent effort will be required to interpret each LM.

4 Techniques

Next, we review the techniques that have been developed to study the fundamental objects described in Section 3 and to perform MI analysis on LMs.

4.1 Logit Lens

The *logit lens* (Figure 4) was first introduced by nostalgebraist (2020), which provides insights on how LMs refine their prediction across layers. This approach has been applied to interpret *activations* in both feature (Geva et al., 2021) and circuit discovery (Lieberum et al., 2023; Wang et al., 2022a). Specifically, the activation h_i^l in each layer l can be projected to the vocabulary space by multiplying it with the unembedding matrix W_U, yielding the logits over the vocabulary $\mathcal{V}$. In doing so, tokens with the highest logits can be used to infer what information is encoded in h_i^l.

The logit lens can also be used to interpret the *weights* of transformer components. For example, Geva et al. (2022) used it to understand the role of the FF sublayer in the LM's prediction by projecting the columns of W_v^l parameter matrix to the vocabulary space. Dar et al. (2023) applied the technique to project query-key ($W_Q^{l,h} W_K^{l,h}$) and value-output ($W_V^{l,h} W_O^l$) interaction parameter matrices to vocabulary space to study how attention heads transfer and mix information from source tokens to the target token.

However, vocabulary space projection may yield seemingly nonsensical results for some models. For instance, the top-1 projected token for BLOOM (Scao et al., 2022) in many intermediate layers is often the input token (Belrose et al., 2023). The authors posited that this may be caused by the discrepancy between the intermediate layers and the last layer (where W_U was trained to apply to). To address it, Belrose et al. (2023); Din et al. (2023); Pal et al. (2023) transformed the intermediate activations to align with the representation space of the final layer.

4.2 Probing

The probing technique (Figure 5) is used extensively in (and also before) MI to investigate whether specific information, such as the part-of-speech linguistic property, is encoded in given intermediate activations (a single neuron or set of neurons) (Conneau et al., 2018; Tenney et al., 2019a,b).

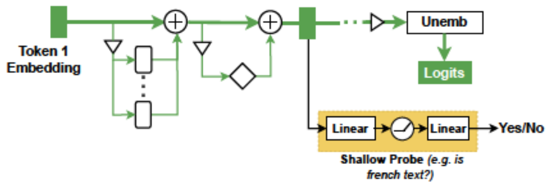


Figure 5: Probing on RS to detect whether it encodes a "French text" feature.

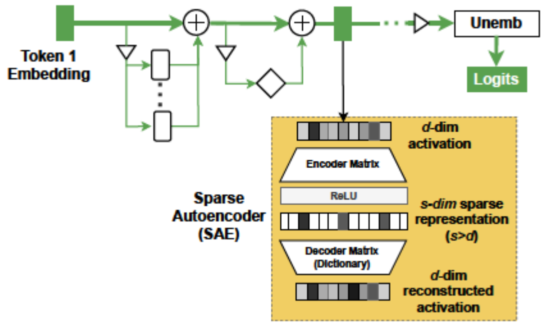


Figure 6: Sparse Autoencoder (SAE) applied to activation on RS.

To do so, it trains a shallow (or linear) classifier, known as a *probe*, to predict whether a feature is present in that activations (Gurnee et al., 2023; Dalvi et al., 2019; Durrani et al., 2020; Antverg and Belinkov, 2021). However, it is important to note that the results of probing analyses only indicate a correlation, not a causal relation, between the feature and activations.

4.3 Sparse Autoencoder (SAE)

SAEs (Figure 6) serve as an unsupervised technique for discovering features from activations, especially those that demonstrate *superposition*, a phenomenon in LMs where their d-dimensional representation encodes more than d features (Elhage et al., 2022b; Sharkey et al., 2023; Cunningham et al., 2023; Bricken et al., 2023; Yun et al., 2021). In contrast to dimensionality reduction techniques (e.g., Principal Component Analysis), SAEs seek to embed the activation vectors into a much higher-dimensional space, but with strong sparsity. Specifically, an encoder maps the d-dimensional input into an s-dimensional vector ($s > d$), which the decoder then maps back to the d-dimension. The encoder and decoder are jointly trained for input reconstruction and sparsity of the s-dimensional representation. This s-dimensional representation, owing to its sparsity, makes the discovery of independent (or *monosemantic*) features more easily.

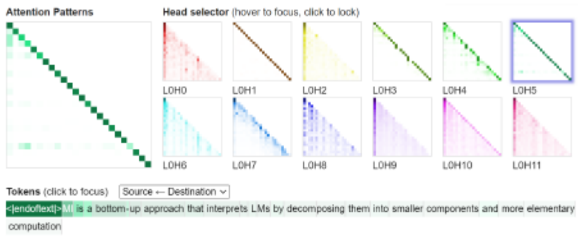


Figure 7: Attention visualization, created using Cooney and Nanda (2023).

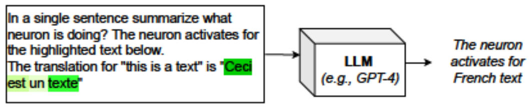


Figure 8: Automated feature explanation, where the highlighted text represents the activation strength for a neuron. An LLM can then be used to generate a textual explanation of the neuron.

4.4 Visualization

Visualization (Figure 7) is employed across various stages of an MI investigation, from generating initial hypotheses to refining them, conducting qualitative analyses, and validating results. For instance, attention patterns are often visualized to understand attention heads (Lieberum et al., 2023; Olsson et al., 2022); a neuron activation across the input text is visualized to identify its functionality (Elhage et al., 2022a; Bricken et al., 2023). While visualization can be highly useful, it requires human effort to interpret results and carries the risks of overgeneralization. Thus, any claims need to be substantiated with further experimentation and analysis.

4.5 Automated Feature Explanation

For feature discovery, human effort is required to annotate neurons with interpretable feature labels based on their activation patterns (Elhage et al., 2022a; Conmy et al., 2024) or the projected tokens when using logits lens (Geva et al., 2022). To reduce human labor, Bills et al. (2023) proposed using LLMs to generate feature labels automatically (Figure 8). Additionally, they also introduced a quantitative automatic explanation score to measure the quality of these explanations, using LLMs to simulate activations based on the automatically generated labels and comparing them with the ground-truth activations.

4.6 Knockout / Ablation

Knockout or *ablation* (Figure 9) is primarily used to identify components in a circuit that are impor-

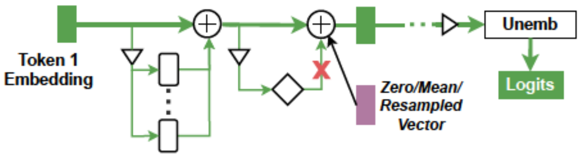


Figure 9: An example of ablation, where the output activation from an FF sublayer is replaced with a zero/mean/resampled vector.

tant to a certain LM behavior. It removes specific components and analyzes the change in the LM behavior, where a significant change suggests importance. There are three main ablation methods: (1) **zero**: replacing the output of the component with a zero vector (Olsson et al., 2022); (2) **mean**: replacing the output with the mean value of randomly sampled inputs from the same input distribution (Wang et al., 2022a); (3) **resampling**: replacing the output with that of another random input (Chan et al., 2022).

Chan et al. (2022) introduced **causal scrubbing** which employs resampling ablation for hypothesis verification. Rather than a direct ablation, they formalize a hypothesis as (G, I, c), where G is the model's computational graph, I is an interpretation graph reflecting the hypothesis, and c is a function that maps between nodes of I to G. Resampling ablation is then performed on G to validate I. The model's behavior should be preserved when one ablates activations in G with random samples that follow the same hypothesis.

4.7 Causal Mediation Analysis (CMA)

Causal mediation analysis (CMA) is popular for circuit discovery, including two main patching approaches (Figure 10).

Activation patching localizes important *components* in a circuit (Vig et al., 2020; Meng et al., 2022), which performs three rounds of model inference. First, the *clean run* uses a prompt that demonstrates the LM capability in investigation. Then, the *corrupt run* uses the same prompt, but with a few important tokens corrupted with noise such that the LM fails to demonstrate the capability.[4] Finally, the *patch run* uses all activation values of the corrupt run, except those of the localized component, which uses the clean run values. The recovery of the lost capability indicates that the patched component is important.

[4]Alternatively, counterfactual prompts can be used as the corruption procedure (Wang et al., 2022a).

Path patching localizes important *connections* between components (Wang et al., 2022a; Goldowsky-Dill et al., 2023). To assess whether the connection between two components, C_1 and C_2, is significant, path patching applies activation patching to the output of C_1 but only along paths serve as the input to C_2. If a change in behavior is observed, then the connection between the two components is considered important.

Circuit discovery via patching is often computationally inefficient, as one has to iteratively patch each component or connection in an LM to measure its importance. To address it, Conmy et al. (2024) proposed ACDC (Automatic Circuit DisCovery) to automate the iterative localization process. Nanda et al. (2022); Kramár et al. (2024) proposed **attribution patching** to approximate activation patching, which requires only two forward passes and one backward pass for measuring all model components. Syed et al. (2023) further extended it to edge attribution patching (EAP), which outperforms ACDC in circuit discovery.

5 Evaluation

Existing work has performed two types of evaluation, i.e., *intrinsic evaluation* that seeks to establish the quality of the explanations on their own, without the context of any particular application, and *extrinsic evaluation*, which measures the quality of the interpretation by applying the obtained insights to a relevant downstream task. This section will focus on the former, whereas the latter was driven by the applications which we will present in Section 7.6.

Faithfulness Faithfulness measures the degree to which an MI interpretation accurately reflects the actual decision-making process underlying a model's behavior under study (Jacovi and Goldberg, 2020). For feature discovery, Elhage et al. (2022a) and Bricken et al. (2023) recruited human annotators to rate the interpretation of a feature based on its activations over texts, and Bills et al. (2023) automated it with GPT-4. For circuit discovery, localization is evaluated by comparing the full model vs. the partial model with the localized circuit alone (Olsson et al., 2022; Wang et al., 2022a; Marks et al., 2024), while the explanation of circuit components is often validated by knocking out the component and manually examining its effect.

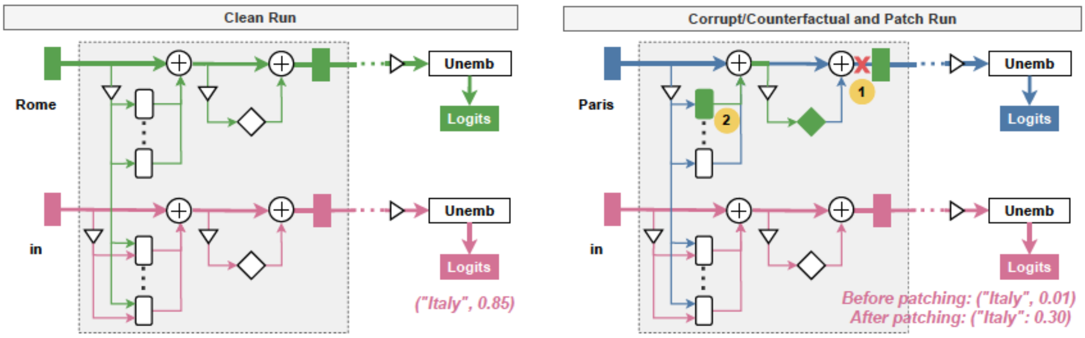


Figure 10: (1) Activation Patching, where in the counterfactual run, an intermediate RS activation is patched (i.e. replaced) by the corresponding activation from the clean run. (2) Path Patching, where all intermediate activations between the green highlighted attention head and FF sublayer are patched by the corresponding activations from the clean run.

Completeness and Minimality Besides faithfulness, completeness and minimality are also desirable (Wang et al., 2022a). For example, when studying the model's ability to recognize French text, an explanation algorithm may identify a group of neurons N or a circuit C. A complete explanation would mean that no other neurons outside of N are responsible (e.g., highly activated), or no other circuits besides C implement this functionality. It can be measured by ablation-based techniques (Wang et al., 2022a; Marks et al., 2024). Specifically, if the model behavior remains unchanged after removing the neuron or circuit, then the explanation is incomplete. On the other hand, minimality measures whether all parts of the explanation (e.g., N or C) are necessary, often by ablating parts of the explanation and computing the change in model behavior (Wang et al., 2022a).

Plausibility Jacovi and Goldberg (2020) defined plausibility as "how convincing the interpretation is to humans". In MI, for example, attributing a model behavior to polysemantic neurons may be less plausible as compared to monosemantic ones (Bricken et al., 2023; Marks et al., 2024), as humans may have trouble understanding the former during the evaluation.

6 A Beginner's Roadmap to MI

A key motivation for this survey is to provide a friendly guide for researchers and developers interested in MI to quickly pick up the field. To this end, we provide a *beginner's roadmap* in Figure 11.

6.1 Feature Study

The study of features can be broadly divided into two categories, i.e., *targeted feature discovery*, which aims to determine if a pre-defined feature (e.g., a "French text" feature) is encoded in the activations of LMs, and *open-ended feature discovery*, which aims to discover all interpretable features in the activations of LMs.

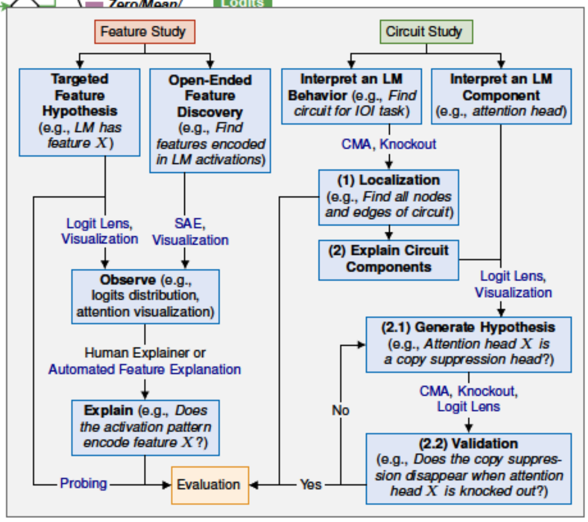


Figure 11: A **Beginner's Roadmap** showing the workflow and common techniques for conducting a feature or circuit study. Both types of studies start with formulating a research question, followed by various steps (enclosed within the blue rectangle) to answer it. Techniques employed to execute each step are indicated along the incoming edge.

Probing, logit lens, and visualization are common techniques used for targeted feature discovery. For probing, a labeled dataset indicating the presence or absence of the target feature is required to train the probe. If the trained probe achieves high accuracy on a held-out test set, it can be inferred that the activation encodes the target feature. On the other hand, the logit lens or visualization requires a human explainer to interpret the logit lens projection or the visualization of the highlighted text. Alternatively, LLMs (e.g., GPT-4) can also be employed to automatically interpret the logit lens projection or highlighted text to confirm if the feature is encoded in the activations. Open-ended feature discovery is typically conducted via SAEs and visualization. Interpretation results of these approaches are then passed to human explainers or LLMs (if going with automatic feature explanation) for annotating them with interpretable feature labels, similar to the annotation process in targeted feature discovery.

Finally, the faithfulness of both categories can be measured by humans based on how interpretable the discovered features are (e.g., "on a scale of 0–3, rate your confidence in this interpretation" as executed by Bricken et al. (2023)) and using automatic explanation scores proposed by Bills et al. (2023).

6.2 Circuit Study

Similarly, the study of circuits can be broadly divided into two categories, i.e., *interpret an LM behavior (e.g., IOI)* and *interpret an LM component (e.g., attention head)*.

The first category involves finding a circuit that is responsible for the LM behavior. To do so, we start with *(1) localization* i.e. identifying all the important components and the edges connecting them via CMA and knockout. Subsequently, the second step is to *(2) explain each component and its connection* one by one. The functional interpretation of each component or edge can be further divided into two sub-steps — *(2.1) Generating a hypothesis:* The behavior of the component under study is analyzed under relevant input using various visualization techniques (e.g., attention visualization) and logit lens vocabulary projection. A human can then generate a plausible hypothesis based on the analysis (e.g., attention head X is hypothesized to function as a copy suppression head, when it was observed to decrease the logit of the token it attends to). *(2.2) Validating the hypothesis:* The hypothesis can be validated using techniques such as knockout, CMA, and logit lens. For instance, if the hypothesis suggests that attention head X is a copy suppression head, then knocking out attention head X should remove the suppression effect. If the hypothesis is validated, the model component is considered interpreted; otherwise, further analysis and new hypotheses are needed. Finally, the discovered circuit, with or without the second-step explanation, can be evaluated in terms of faithfulness, completeness, and minimality (Section 5).

The second category of the circuit study, i.e., interpreting an LM component, follows similar steps but starts with the second step of the first category, given that the target component to interpret has been provided.

7 Findings and Applications

7.1 Findings on Features

Monosemantics vs. Polysemantics Earlier work studied neurons as a natural candidate for encoding features, leading to the discovery of "sentiment neurons" (Radford et al., 2017), "knowledge neurons" (Dai et al., 2022), "Base64 neurons" (Elhage et al., 2022a), "skill neurons" (Wang et al., 2022b), "positional neurons" (Voita et al., 2023), etc. However, many of these studies found that neurons within LMs are not *monosemantic*, i.e. they do not activate only for a single feature. In contrast, they are *polysemantic*, i.e. they activate in response to multiple unrelated features (Elhage et al., 2022a; Gurnee et al., 2023; Elhage et al., 2022b). For instance, a single neuron could activate for both French texts and texts encoded in Base64. Due to its polysemantic nature, mapping a neuron to a specific feature becomes challenging.

Superposition The discovery of polysemantic neurons has led to the hypothesis of *superposition*, i.e., a model can represent a greater number of independent features than the number of available neurons. Specifically, features represented in superposition are encoded in activations by a linear combination of multiple neurons, rather than by a single neuron, at the cost of some interference with each other. This hypothesis has been verified by Elhage et al. (2022b), where the authors showed that when features are sparse, the model tends to encode features in activation space using superposition. To extract features from such representations, SAEs have become a widely adopted approach (Bricken et al., 2023; Sharkey et al., 2023; Riggs, 2023; Cunningham et al., 2023), as discussed in Section 4.3. Early studies on SAEs have shown promising results, with the extracted features from SAEs appearing more interpretable than those from neurons, according to both human analysis and automatic explanation scores (Bricken et al., 2023).

7.2 Findings on Circuits

Interpreting LM Behaviors Specific circuits have been identified for various LM behaviors such as in-context learning (Olsson et al., 2022), indirect object identification (IOI) (Wang et al., 2022a), greater-than operations (Conmy et al., 2024), Python docstring formatting (Heimersheim and Janiak, 2023), modular addition (Nanda et al., 2023a; Zhong et al., 2024), and more (Lieberum et al., 2023; Marks et al., 2024). These circuits are defined at varying levels of granularity, where nodes in the computation graph are defined as the outputs of MHA and FF sublayers (Olsson et al., 2022; Wang et al., 2022a) or as SAE features (Marks et al., 2024; Cunningham et al., 2023). In addition, Merullo et al. and Quirke et al. (2024) showed that the same components (e.g., induction heads) are reused by different circuits (e.g., IOI and induction circuits) to implement different tasks. Finally, to demonstrate the scalability of current techniques for identifying circuits in LLMs, Lieberum et al. (2023) identified the circuit used for the multiple-choice question-answering task on the 70B Chinchilla LLM (Hoffmann et al., 2022)

Interpreting Transformer Components The study of circuits has also yielded insights into the functionalities of transformer components.

The RS can be viewed as a one-way communication channel that transfers information from earlier to later layers. Furthermore, Elhage et al. (2021) hypothesized that MHA and FF in different layers write their output in different subspaces of the RS, which prevents interference of information. In addition, nostalgebraist (2020) proposed to view the RS as an LM's current "guess" for the output, which is iteratively refined layer-by-layer..

The MHA sublayers are responsible for moving information between tokens, which enables information from other tokens (i.e., context) to be incorporated into each token's representation. Elhage et al. (2021) showed that each attention head in a layer operates independently and can be interpreted independently. Several MI studies have shown that these attention heads seem to have specialized roles. For instance, "negative heads" discovered in GPT2-small by McDougall et al. (2023) are responsible for reducing the logit values of the tokens that have already appeared in the context. Other notably identified attention heads include previous token heads (Wang et al., 2022a), duplicate token heads (Wang et al., 2022a), copying heads (Elhage et al., 2021), induction heads (Olsson et al., 2022), and successor heads (Gould et al., 2023).

The FF sublayers are attributed for the majority of feature extraction (Gurnee et al., 2023), storing and recalling pre-trained knowledge (Meng et al., 2022) and arithmetic computation (Stolfo et al., 2023a). More generally, Geva et al. (2021) viewed FF sublayers as key-value stores where the outputs of the first layer (W_k^l) of the FF sublayer serves as keys that activate values (stored knowledge) within the weight matrices of the second layer (W_v^l). Furthermore, they demonstrated that earlier FF layers typically process shallow (syntactic or grammatical) input patterns, while later layers focus more on semantic patterns (e.g., text related to TV shows).

7.3 Findings on Universality

Studies on universality have yielded mixed results. Early circuit analyses identified components such as induction heads (Olsson et al., 2022), successor heads (Gould et al., 2023), and duplication heads (Wang et al., 2022a) across multiple LMs. Similarly, Merullo et al. found that different circuits implementing different tasks (IOI and colored objects task) reuse the same components (e.g., induction head), demonstrating universality across tasks. However, Zhong et al. (2024) discovered that two LMs trained with different initialization can develop qualitatively different circuits for the modular addition task. Similarly, Chughtai et al. (2023) found that LMs trained to perform group composition on finite groups with different random weight initializations on the same task do not develop similar representations and circuits. Finally, Gurnee et al. (2024) found that only about 1-5% of neurons from GPT-2 models trained with random initialization exhibit universality. Understanding the degrees of feature and circuit universality and their dependency on various aspects of model training (e.g., initialization, model size, and loss function) remains a crucial open problem.

7.4 Findings on Model Capabilities

In-Context Learning (ICL) ICL is an emergent ability of LLMs that enables them to adapt to new tasks based solely on instructions or a few demonstrations at inference time (Wei et al., 2022). Elhage et al. (2021) studied a simplified case of ICL and discovered an induction circuit composed of attention heads with specialized roles (e.g., induction heads), which were then found to be crucial even for general cases of ICL (Olsson et al., 2022; Bansal et al., 2023). Ren et al. (2024) further investigated few-shot ICL and identified "semantic induction heads", which, unlike prior induction heads, model the semantic relationship between the input and the output token (e.g., “I have a nice pen for writing. The pen is nice to” → “write”).

Reasoning Stolfo et al. (2023b) studied arithmetic reasoning and found that attention heads are responsible for transferring information from operand and operator tokens to the RS of the answer or output token, with FF modules subsequently calculating the answer token. Dutta et al. (2024) studied chain-of-thought (CoT) multi-step reasoning over fictional ontologies and found that LLMs seem to deploy multiple different neural pathways in parallel to compute the final answer. Concurrently, Rai and Yao (2024) investigated neuron activation (Geva et al., 2022) as a unified lens to explain how CoT elicits arithmetic reasoning of LLMs, including phenomena that were only empirically discussed in prior work (Wang et al., 2023; Ye et al., 2023; Madaan and Yazdanbakhsh, 2022). In particular, the authors found neurons representing various arithmetic reasoning concepts and showed that the activation of these neurons is necessary but not sufficient for an LLM’s arithmetic reasoning. Finally, Brinkmann et al. (2024) discovered an interpretable algorithm in LM for the task of pathfinding in trees.

Others As listed in Section 7.2, prior work has also studied LM capabilities in tasks such as IOI (Wang et al., 2022a), modular addition (Nanda et al., 2023b), and greater-than operations (Conmy et al., 2024), leading to the discovery of circuits that implement these tasks. Compared with the interpretation of ICL and reasoning, these studies not only justified the rationale of a capability but also revealed its underlying algorithm through circuits.

7.5 Findings on Learning Dynamics

Phase Changes during LM Training Prior studies have observed sudden shifts in LMs’ capabilities, called “phase changes” (Olsson et al., 2022; Power et al., 2022; Wei et al., 2022). These changes are considered key steps during the LM training. MI has been applied to examine the relationship between the emergence of features and circuits and these phase changes. For example, Olsson et al. (2022) found correlations between phase changes and the formation of induction circuits, suggesting that the development of these circuits underlies the phase change. In the task of symbol manipulation, Nanda et al. (2023a); Varma et al. (2023) discovered a similar correlation contributing to LM grokking (Power et al., 2022), a phenomenon of LM generalizing beyond memorization. Chen et al. (2024) found that sudden drops in the loss during training correspond to the acquisition of attention heads that recognize specific syntactic relations. Finally, Huang et al. (2024b) provided a unified explanation for grokking, double descent (Nakkiran et al., 2021), and emergent abilities (Wei et al., 2022) as a competition between memorization and generalization circuits.

Learning Dynamics during LM Fine-Tuning Prakash et al. (2024) investigated the underlying changes in mechanisms (e.g., task-relevant circuits) to understand performance enhancements in fine-tuned LMs. The authors found that fine-tuning does not fundamentally change the mechanisms but enhances existing ones.

7.6 Applications of MI

7.6.1 Model Enhancement

Knowledge Editing LMs are known to store factual knowledge encountered during pre-training (Petroni et al., 2019; Cohen et al., 2023). For instance, when an LM is prompted with “The space needle is in the city of”, it may retrieve the stored facts and correctly predict “Seattle”. However, these stored facts may be incorrect or outdated over time, leading to factually incorrect generation (Cohen et al., 2024). MI has been found to be a helpful tool for addressing the problem, including understanding where and how facts are stored within LMs, how they are recalled during the inference time, and the approaches for knowledge editing (Meng et al., 2022, 2023; Geva et al., 2023; Sharma et al., 2024). For instance, Meng et al. (2022) used path patching to localize components that are responsible for storing factual knowledge, and then edited the fact (e.g., replacing “Seattle” with “Paris”) by only updating the parameters of those components.

LM Generation Steering LM generation steering involves controlling an LM’s output by manipulating its activations at inference time. For instance, Geva et al. (2022) proposed a method to suppress toxic language generation by identifying and manually activating neurons in FF layers responsible for promoting non-toxic or safe words. Similarly, Templeton et al. (2024) identified safety-related features (e.g., unsafe code, gender bias) and manipulated their activations to steer the LM towards (un)desired behaviors (e.g., safe code generation, unbiased text generation). Additionally, Nanda et al. (2023b) demonstrated that an LM's output can be altered (e.g., flipping a player turn in the game of Othello from YOURS to MINE) by pushing its activation in the direction of a linear vector representing the desired behavior, which was identified using a linear probe.

7.6.2 AI Safety

AI safety is an important concern that MI aims to address. At present, the exact role MI can play in addressing AI safety is unclear (Casper, 2023). Within MI, "enumerative safety" aims to address AI safety by enumerating all features in LMs and inspecting those related to dangerous capabilities or intentions (Elhage et al., 2022b; Olah and Jermyn, 2023). To this end, Templeton et al. (2024) identified several safety-relevant features that not only activate when the LM exhibits specific behaviors but also causally influence the LM's output; however, the specific circuits that use these features to implement the behavior have not yet been identified. As we have discussed, Geva et al. (2022) encouraged language safety by steering the LM's generation of non-toxic tokens. Finally, insights from circuit studies were used to detect prompt injection (Belrose et al., 2023) and find adversarial examples for the IOI task (Wang et al., 2022a).

7.6.3 Others

Insights from MI have also been used for other downstream tasks. Marks et al. (2024) improved the generalization of classifiers by identifying and ablating spurious features that humans consider to be task-irrelevant. Geva et al. (2022) proposed self-supervised early exit prediction for efficient inference, drawing insights from their investigations of the FF sublayers' role in token prediction.

8 Discussion and Future Work

Automated Hypothesis Generation At a high level, MI study is a process with two stages: generating hypotheses on the underlying mechanisms of LM behavior and validating them. Although various techniques have automated hypothesis validation (Section 4.5), the generation part is mainly left to humans, a potential scalability bottleneck to LLMs and complex behaviors. To address it, automated hypothesis generation, potentially with humans in the loop, is instrumental.

Studies on Complex Tasks and LLMs Current MI studies are mostly performed on simpler tasks, often criticized as "streetlight interpretability" (Casper, 2023; Wang, 2022). For instance, Wang (2022) intentionally selected the IOI task (Wang et al., 2022a) because it is a simple algorithmic task. Similarly, although a few studies were done on "production-level" LMs (Lieberum et al., 2023; Templeton et al., 2024), most still used small LMs, which may have limited generalizability, given the mixed results on universality.

Practical Utility While people acknowledge the importance of actionable insights to downstream applications (Doshi-Velez and Kim, 2017), MI studies that highlight such applications, such as adversarial example generation (Wang et al., 2022a), often lack the depth of investigation when they do not compare against alternative methods or baselines. As an exception, Marks et al. (2024) showed that their MI technique improves model generalization better than several baselines.

Standardized Benchmarks and Metrics Evaluating interpretability results is inherently challenging due to the lack of ground truth (Zhou et al., 2022), and current MI studies also employ diverse *ad hoc* evaluation approaches, potentially leading to inconsistent comparisons (Zhang and Nanda, 2023). There are some proposals for standardized evaluation benchmarks. RAVEL (Huang et al., 2024a) evaluates techniques (e.g., SAEs) that disentangle polysemantic neurons into monosemantic features. Tracr (Lindner et al., 2024) converts human-readable RASP programs (Weiss et al., 2021) into transformers to enable ground-truth features and algorithms. However, these proposed benchmarks are still insufficient (Räuker et al., 2023). For instance, success on synthetic transformers in the Tracr benchmark may not imply that on naturally trained transformers. Thus, more effort is needed to develop standard evaluation techniques and to ensure their wide adoption.

9 Limitations

Our survey paper discusses MI studies conducted only on decoder-only transformer LMs, omitting studies on other transformer LM variants such as encoder-only and encoder-decoder transformer LMs. Additionally, due to space constraints, some techniques may not have been presented with full technical details (e.g., causal scrubbing).

Furthermore, given the fast-paced nature of the field, our survey may not reflect the latest advancements that were published close to or after the survey was conducted. Despite these limitations, however, our survey provides a comprehensive taxonomy of the current state of the MI field as well as a roadmap, specifically curated to help newcomers quickly pick up the field. Both the taxonomy and the roadmap can serve as guides for future researchers interested in encoder-only or encoder-decoder LMs to build up their survey. In addition, we have created a GitHub repository at https://github.com/Dakingrai/awesome-mechanistic-interpretability-lm-papers to sustainably maintain this survey, and we welcome researchers sharing the same interest to add new papers and extend the taxonomy and the roadmap.

Acknowledgements

DR and ZY were sponsored by the National Science Foundation (SHF2311468) and the College of Computing and Engineering and the Department of Computer Science at George Mason University. This project was also supported by resources provided by the Office of Research Computing at George Mason University (https://orc.gmu.edu) and funded in part by grants from the National Science Foundation (2018631).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Omer Antverg and Yonatan Belinkov. 2021. On the pitfalls of analyzing individual neurons in language models. In *International Conference on Learning Representations*.

Hritik Bansal, Karthik Gopalakrishnan, Saket Dingliwal, Sravan Bodapati, Katrin Kirchhoff, and Dan Roth. 2023. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11833–11856.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Leonard Bereska and Efstratios Gavves. 2024. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. https://openai.com/index/language-models-can-explain-neurons-in-language-models/.

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

Jannik Brinkmann, Abhay Sheshadri, Victor Levoso, Paul Swoboda, and Christian Bartelt. 2024. A mechanistic analysis of a transformer trained on a symbolic multi-step reasoning task. *arXiv preprint arXiv:2402.11917*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Stephen Casper. 2023. The engineer's interpretability sequence. https://www.lesswrong.com/s/a6ne2ve5uturEEQK7.

Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldwosky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. 2022. Causal scrubbing, a method for rigorously testing interpretability hypotheses. *AI Alignment Forum*. https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN/causal-scrubbing-a-method-for-rigorously-testing.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Angelica Chen, Ravid Schwartz-Ziv, Kyunghyun Cho, Matthew L. Leavitt, and Naomi Saphra. 2024. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms. In *The Twelfth International Conference on Learning Representations*.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. 2023. A toy model of universality: Reverse engineering how networks learn group operations. In *International Conference on Machine Learning*, pages 6243–6267. PMLR.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Roi Cohen, Mor Geva, Jonathan Berant, and Amir Globerson. 2023. Crawling the internal knowledge-base of language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1856–1869.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2024. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Alan Cooney and Neel Nanda. 2023. Circuitsvis. https://github.com/TransformerLensOrg/CircuitsVis.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 2019. 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*, volume 33, pages 6309–6317.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. Analyzing transformers in embedding space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16124–16170.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2023. Jump to conclusions: Short-cutting transformers with linear transformations. *arXiv preprint arXiv:2303.09435*.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880.

Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. 2024. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *arXiv preprint arXiv:2402.18312*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Andy Jones, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. 2022a. Softmax linear units. *Transformer Circuits Thread*. https://transformer-circuits.pub/2022/solu/index.html.

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

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. https://transformer-circuits.pub/2021/framework/index.html.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*.

Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. 2023. Successor heads: Recurring, interpretable attention heads in the wild. In *The Twelfth International Conference on Learning Representations*.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. 2024. Universal neurons in gpt2 language models. *arXiv preprint arXiv:2401.12181*.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*.

Stefan Heimersheim and Jett Janiak. 2023. A circuit for python docstrings in a 4-layer attention-only transformer. https://www.alignmentforum.org/posts/u6KXXmKFbXfWzoAXn/acircuit-for-python-docstrings-in-a-4-layer-attention-only.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. 2024a. Ravel: Evaluating interpretability methods on disentangling language model representations. *arXiv preprint arXiv:2402.17700*.

Yufei Huang, Shengding Hu, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024b. Unified view of grokking, double descent and emergent abilities: A perspective from circuits competition. *arXiv preprint arXiv:2402.15175*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. 2024. Atp*: An efficient and scalable method for localizing llm behaviour to components. *arXiv preprint arXiv:2403.00745*.

Tom Lieberum, Matthew Rahtz, János Kramár, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*.

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Tom McGrath, and Vladimir Mikulik. 2024. Tracr: Compiled transformers as a laboratory for interpretability. *Advances in Neural Information Processing Systems*, 36.

Aman Madaan and Amir Yazdanbakhsh. 2022. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*.

Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2023. Copy suppression: Comprehensively understanding an attention head. *arXiv preprint arXiv:2310.04625*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Circuit component reuse across tasks in transformer language models. In *The Twelfth International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2021. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023a. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023b. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*.

Neel Nanda, Chris Olah, Catherine Olsson, Nelson Elhage, and Hume Tristan. 2022. Attribution patching: Activation patching at industrial scale, 2023. https://www.neelnanda.io/mechanistic-interpretability/attribution-patching.

nostalgebraist. 2020. Interpreting gpt: the logit lens. *AI Alignment Forum*. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. https://distill.pub/2020/circuits/zoom-in.

Chris Olah and Adam Jermyn. 2023. What would be the most safety-relevant features in language models? (circuits updates - july 2023). *Transformer Circuits Thread*. https://transformer-circuits.pub/2023/july-update/index.html#safety-features.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C Wallace, and David Bau. 2023. Future lens: Anticipating subsequent tokens from a single hidden state. *arXiv preprint arXiv:2311.04897*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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)*, pages 2463–2473.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. 2022. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *The Twelfth International Conference on Learning Representations*.

Philip Quirke, Clement Neo, and Fazl Barez. 2024. Increasing trust in language models through the reuse of verified circuits. *arXiv preprint arXiv:2402.02619*.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Daking Rai and Ziyu Yao. 2024. An investigation of neuron activation as a unified lens to explain chain-of-thought eliciting arithmetic reasoning of llms. *arXiv preprint arXiv:2406.12288*.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. 2023. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 464–483. IEEE.

Jie Ren, Qipeng Guo, Hang Yan, Dongrui Liu, Xipeng Qiu, and Dahua Lin. 2024. Identifying semantic induction heads to understand in-context learning. *arXiv preprint arXiv:2402.13055*.

Logan Riggs. 2023. Found 600+ monosemantic features in a small lm using sparse autoencoders. https://www.lesswrong.com/posts/wqRqb7h6ZC48iDgfK/tentatively-found-600-monosemantic-features-in-a-small-lm.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Lee Sharkey, Dan Braun, and Beren Millidge. 2023. Taking features out of superposition with sparse autoencoders, 2022. https://www.alignmentforum.org/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition.

Arnab Sen Sharma, David Atkinson, and David Bau. 2024. Locating and editing factual associations in mamba. *arXiv preprint arXiv:2404.03646*.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023a. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023b. Understanding arithmetic reasoning in language models using causal mediation analysis. *arXiv preprint arXiv:2305.15054*.

Aaquib Syed, Can Rager, and Arthur Conmy. 2023. Attribution patching outperforms automated circuit discovery. *arXiv preprint arXiv:2310.10348*.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*. https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. 2023. Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2023. Neurons in large language models: Dead, n-gram, positional. *arXiv preprint arXiv:2309.04827*.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Kevin Wang. 2022. Gears-level mental models of transformer interpretability. https://www.lesswrong.com/posts/X26ksz4p3wSyycKNB/gears-level-mental-models-of-transformer-interpretability.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022a. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. In *The Eleventh International Conference on Learning Representations*.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022b. Finding skill neurons in pre-trained transformer-based language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11132–11152.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2021. Thinking like transformers. In *International Conference on Machine Learning*, pages 11080–11090. PMLR.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Veselin Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2023. Complementary explanations for effective in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4469–4484, Toronto, Canada. Association for Computational Linguistics.

Zeyu Yun, Yubei Chen, Bruno Olshausen, and Yann Lecun. 2021. 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*, pages 1–10.

Fred Zhang and Neel Nanda. 2023. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. 2024. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *Advances in Neural Information Processing Systems*, 36.

Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. 2022. Do feature attribution methods correctly attribute features? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9623–9633.