

A PRIMER ON THE INNER WORKINGS OF TRANSFORMER-BASED LANGUAGE MODELS

Javier Ferrando^{1*}, Gabriele Sarti², Arianna Bisazza², Marta R. Costa-jussà³

¹Universitat Politècnica de Catalunya, ²CLCG, University of Groningen, ³FAIR, Meta

ABSTRACT

The rapid progress of research aimed at interpreting the inner workings of advanced language models has highlighted a need for contextualizing the insights gained from years of work in this area. This primer provides a concise technical introduction to the current techniques used to interpret the inner workings of Transformer-based language models, focusing on the generative decoder-only architecture. We conclude by presenting a comprehensive overview of the known internal mechanisms implemented by these models, uncovering connections across popular approaches and active research directions in this area.

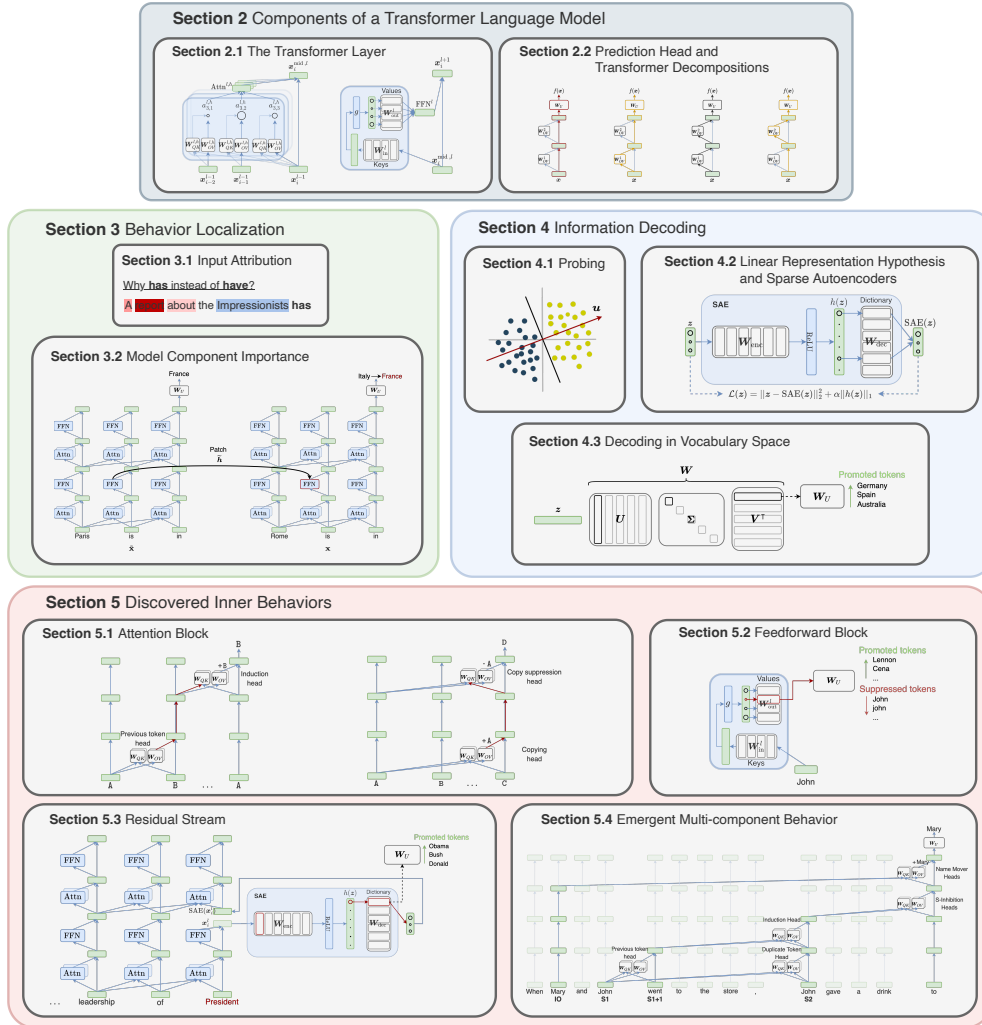


Figure 1: Survey overview. **Section 2** introduces the Transformer language model and its components. **Section 3** and **Section 4** present interpretability techniques used to analyze models' inner workings. Finally, **Section 5** presents known inner workings of Transformer language models.

*Correspondence to: jferrandomonsonis@gmail.com.

1 INTRODUCTION

The development of powerful Transformer-based language models (LMs; Radford et al., 2019; Brown et al., 2020; Hoffmann et al., 2022; Chowdhery et al., 2023) and their widespread utilization underscores the significance of research devoted to understanding their inner mechanisms. Gaining a deeper understanding of these mechanisms in highly capable AI systems holds important implications in ensuring the safety and fairness of such systems, mitigating their biases and errors in critical settings, and ultimately driving model improvements (Wei et al., 2022; Costa-jussà et al., 2023). As a result, the natural language processing (NLP) community has witnessed a notable increase in research focused on interpretability in language models, leading to new insights into their internal functioning.

Existing surveys present a wide variety of techniques adopted by Explainable AI analyses (Räuker et al., 2023) and their applications in NLP (Madsen et al., 2022; Lyu et al., 2024). While previous NLP interpretability surveys primarily focused on encoder-based models like BERT (Devlin et al., 2019; Rogers et al., 2021), the success of decoder-only Transformers (Radford et al., 2018) prompted further developments in the analysis of these powerful generative models, with concurrent work surveying trends in interpretability research and their relation to AI safety (Bereska & Gavves, 2024). By contrast, this work provides a concise, in-depth technical introduction to relevant techniques used in LM interpretability research, focusing on insights derived from models’ inner workings and drawing connections between different areas of interpretability research. Moreover, throughout this work, we employ a unified notation to introduce model components, interpretability methods, and insights from surveyed works, shedding light on the assumptions and motivations behind specific method designs. We categorize LM interpretability approaches surveyed in this work along two dimensions: i) *localizing* the inputs or model components responsible for a particular prediction (Section 3); and ii) *decoding* information stored in learned representations¹ to understand its usage across network components (Section 4). Finally, Section 5 provides an exhaustive list of insights into the inner workings of Transformer-based LMs, and Section 6 provides an overview of useful tools to conduct interpretability analyses on these models.

2 THE COMPONENTS OF A TRANSFORMER LANGUAGE MODEL

Auto-regressive language models assign probabilities to sequences of tokens. Using the probability chain rule, we can decompose the probability distribution over a sequence $\mathbf{t} = \langle t_1, t_2, \dots, t_n \rangle$ into a product of conditional distributions:

$$P(t_1, \dots, t_n) = P(t_1) \prod_{i=1}^n P(t_{i+1} | t_1, \dots, t_i). \quad (1)$$

Such distributions can be parametrized using a neural network optimized to maximize the likelihood of a corpus used for training (Bengio et al., 2003). In recent years, the Transformer architecture by Vaswani et al. (2017) was widely adopted for this purpose thanks to its expressivity and its scalability (Kaplan et al., 2020). While several variants of the original Transformers were proposed, we focus here on the decoder-only architecture (also known as *GPT-like*) due to its success and popularity.² A decoder-only model f has L layers, and operates on a sequence of embeddings $\mathbf{x} = \langle \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \rangle$ representing the tokens $\mathbf{t} = \langle t_1, t_2, \dots, t_n \rangle$. Each embedding $\mathbf{x} \in \mathbb{R}^d$ is a *row vector* corresponding to a row of the embedding matrix $\mathbf{W}_E \in \mathbb{R}^{|\mathcal{V}| \times d}$, where \mathcal{V} is the model vocabulary. Intermediate layer representations, for instance, at position i and layer l , are referred to as \mathbf{x}_i^l .³ By $\mathbf{X} \in \mathbb{R}^{n \times d}$ we represent the sequence \mathbf{x} as a matrix with embeddings stacked as rows. Likewise, for intermediate representations, $\mathbf{X}_{\leq i}^l$ is the layer l representation matrix up to position i . Appendix A provides a summary of the notation used in this work.

Following recent literature regarding interpretability in Transformers, we present the architecture adopting the *residual stream* perspective (Elhage et al., 2021a). In this view, each input embedding gets updated via vector additions from the attention (Section 2.1.2) and feed-forward

¹In this work we use *representations* and *activations* interchangeably, and we refer to the fundamental unit of information encoded in model activations as *features*, representing human-interpretable input properties.

²Most of the insights presented in this work remain relevant for encoder-only and encoder-decoder models.

³Note that $\mathbf{x}_i^0 = \mathbf{x}_i$.

blocks (Section 2.1.3), producing *residual stream states* (or intermediate representations). The final layer residual stream state is then projected into the vocabulary space via the unembedding matrix $\mathbf{W}_U \in \mathbb{R}^{d \times |\mathcal{V}|}$ (Section 2.2), and normalized via the softmax function to obtain the probability distribution over the vocabulary from which a new token is sampled.

2.1 THE TRANSFORMER LAYER

In this section, we present the Transformer layer components following their computations' flow.

2.1.1 LAYER NORMALIZATION


Layer normalization (LayerNorm) is a common operation used to stabilize the training process of deep neural networks (Ba et al., 2016). Although early Transformer models implemented LayerNorm at the output of each block, modern models consistently normalize preceding each block (Xiong et al., 2020; Takase et al., 2023). Given a representation \mathbf{z} , the LayerNorm computes $(\mathbf{z} - \mu(\mathbf{z})/\sigma(\mathbf{z})) \odot \gamma + \beta$, where μ and σ calculate the mean and standard deviation, and $\gamma \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^d$ refer to learned element-wise transformation and bias respectively. Layer normalization can be interpreted geometrically by visualizing the mean subtraction operation as a projection of input representations onto a hyperplane defined by the normal vector $[1, 1, \dots, 1] \in \mathbb{R}^d$, and the following scaling to \sqrt{d} norm as a mapping of the resulting representations to a hypersphere (Brody et al., 2023). Kobayashi et al. (2021) notes that LayerNorm can be treated as an affine transformation $\mathbf{z}\mathbf{L} + \beta$, as long as $\sigma(\mathbf{z})$ is considered as a constant (Appendix B). In this view, the matrix \mathbf{L} computes the centering and scaling operations. Furthermore, the weights of the affine transformation can be folded into the following linear layer (Appendix C), simplifying the analysis.

We note that current LMs such as Llama 2 (Touvron et al., 2023) adopt an alternative layer normalization procedure, RMSNorm (Zhang & Sennrich, 2019), where the centering operation is removed, and scaling is performed using the root mean square (RMS) statistic.

2.1.2 ATTENTION BLOCK

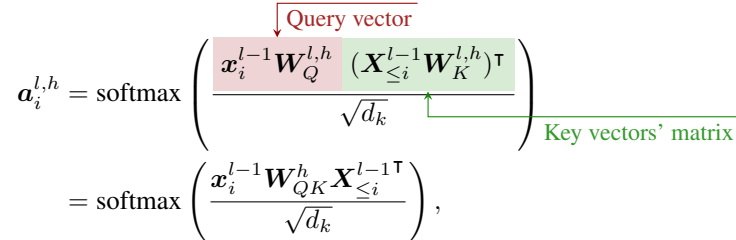
Attention is a key mechanism that allows Transformers to contextualize token representations at each layer. The attention block is composed of multiple *attention heads*. At a decoding step i , each attention head reads from residual streams across previous positions ($\leq i$), decides which positions to attend to, gathers information from those, and finally writes it to the current residual stream. We adopt the rearrangement proposed by Kobayashi et al. (2021) and Elhage et al. (2021a) to simplify the analysis of residual stream contributions.⁴ In particular, every attention head computes

$$\begin{aligned} \text{Attn}^{l,h}(\mathbf{X}_{\leq i}^{l-1}) &= \sum_{j \leq i} a_{i,j}^{l,h} \mathbf{x}_j^{l-1} \mathbf{W}_V^{l,h} \mathbf{W}_O^{l,h} \\ &= \sum_{j \leq i} a_{i,j}^{l,h} \mathbf{x}_j^{l-1} \mathbf{W}_{OV}^{l,h}, \end{aligned} \quad (2)$$



where the learnable weight matrices $\mathbf{W}_V^{l,h} \in \mathbb{R}^{d \times d_h}$ and $\mathbf{W}_O^{l,h} \in \mathbb{R}^{d_h \times d}$ are combined into the OV matrix $\mathbf{W}_V^{l,h} \mathbf{W}_O^{l,h} = \mathbf{W}_{OV}^{l,h} \in \mathbb{R}^{d \times d}$, also referred to as *OV (output-value) circuit*. The attention weights for every key ($\leq i$) given the current query (i) are obtained as:

$$\begin{aligned} a_i^{l,h} &= \text{softmax} \left(\frac{\mathbf{x}_i^{l-1} \mathbf{W}_Q^{l,h} (\mathbf{X}_{\leq i}^{l-1} \mathbf{W}_K^{l,h})^\top}{\sqrt{d_k}} \right) \\ &= \text{softmax} \left(\frac{\mathbf{x}_i^{l-1} \mathbf{W}_{QK}^{l,h} \mathbf{X}_{\leq i}^{l-1 \top}}{\sqrt{d_k}} \right), \end{aligned} \quad (3)$$



⁴The original implementation considers a concatenation of each attention head output before projecting into the weight matrix $\mathbf{W}_O^l \in \mathbb{R}^{H \cdot d_h \times d}$. By splitting \mathbf{W}_O^l into per-head weight matrices $\mathbf{W}_O^{l,h} \in \mathbb{R}^{d_h \times d}$, matrices $\mathbf{W}_V^{l,h}$ and $\mathbf{W}_O^{l,h}$ can be joined in a single matrix $\mathbf{W}_{OV}^{l,h}$.

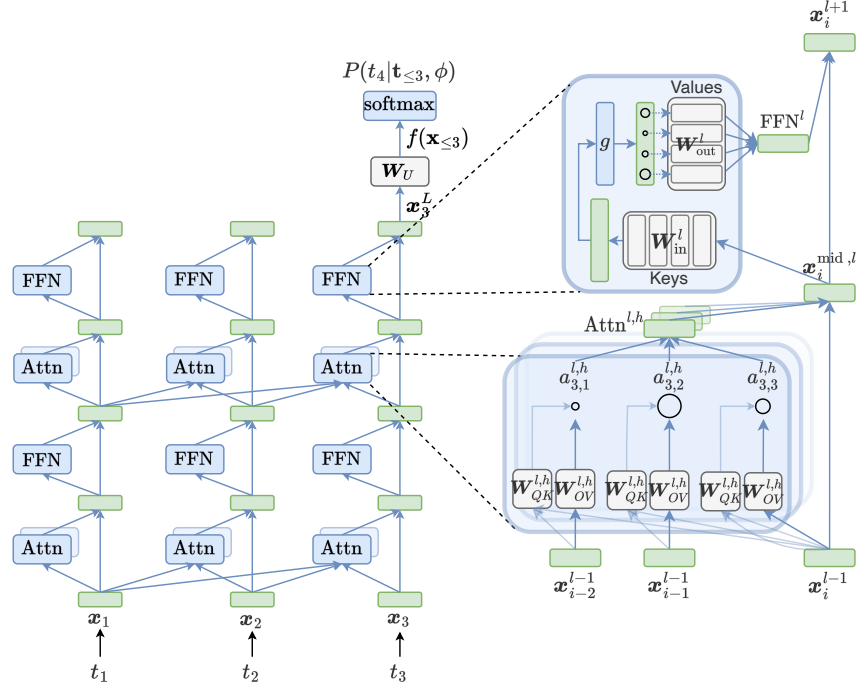


Figure 2: Unrolled Transformer LM with expanded views of the Attention and Feedforward network blocks, including model weights (gray) and residual stream states (green). Based on figures from (Ferrando & Voita, 2024; Voita et al., 2023).

with $\mathbf{W}_Q^{l,h} \in \mathbb{R}^{d \times d_h}$ and $\mathbf{W}_K^{l,h} \in \mathbb{R}^{d \times d_h}$ combining as the *QK (query-key) circuit* $\mathbf{W}_Q^h \mathbf{W}_K^{h\top} = \mathbf{W}_{QK}^h \in \mathbb{R}^{d \times d}$. The decomposition introduced in Equations (2) and (3) enables a view of QK and OV circuits as units responsible for respectively reading from and writing to the residual stream. The attention block output is the sum of individual attention heads, which is subsequently added back into the residual stream:

$$\text{Attn}^l(\mathbf{X}_{\leq i}^{l-1}) = \sum_{h=1}^H \text{Attn}^{l,h}(\mathbf{X}_{\leq i}^{l-1}), \quad (4)$$

$$\mathbf{x}_i^{\text{mid},l} = \mathbf{x}_i^{l-1} + \text{Attn}^l(\mathbf{X}_{\leq i}^{l-1}). \quad (5)$$

2.1.3 FEEDFORWARD NETWORK BLOCK

The feedforward network (FFN) in the Transformer block is composed of two learnable weight matrices⁵: $\mathbf{W}_{\text{in}}^l \in \mathbb{R}^{d \times d_{\text{in}}}$ and $\mathbf{W}_{\text{out}}^l \in \mathbb{R}^{d_{\text{in}} \times d}$. \mathbf{W}_{in}^l reads from the residual stream state $\mathbf{x}_i^{\text{mid},l}$, and its result is passed through an element-wise non-linear activation function g , producing the *neuron activations*. These get transformed by $\mathbf{W}_{\text{out}}^l$ to produce the output $\text{FFN}(\mathbf{x}_i^{\text{mid}})$, which is then added back to the residual stream:

$$\text{FFN}^l(\mathbf{x}_i^{\text{mid},l}) = g(\mathbf{x}_i^{\text{mid},l} \mathbf{W}_{\text{in}}^l) \mathbf{W}_{\text{out}}^l. \quad (6)$$

$$\mathbf{x}_i^l = \mathbf{x}_i^{\text{mid},l} + \text{FFN}^l(\mathbf{x}_i^{\text{mid},l}). \quad (7)$$

The computation described in Equation (6) was equated to *key-value memory retrieval* (Geva et al., 2021), with keys (\mathbf{w}_{in}^l) stored in columns of \mathbf{W}_{in}^l acting as pattern detectors over the input sequence (Figure 2 right) and values $\mathbf{w}_{\text{out}}^l$, rows of $\mathbf{W}_{\text{out}}^l$, being upweighted by each neuron activation. We

⁵We omit bias terms, following the practice of recent models such as Llama (Touvron et al., 2023), PaLM (Chowdhery et al., 2023) and OLMo (Groeneveld et al., 2024), which also exclude biases from attention matrices.

use the term “neuron” to refer to each value after an element-wise non-linearity, and use “unit” or “dimension” for other individual values in any other representation. Provided that the output of the FFN is a linear combination of $\mathbf{w}_{\text{out}}^l$ values, Equation (6) can be rewritten following the key-value perspective:

$$\text{FFN}^l(\mathbf{x}_i^{\text{mid},l}) = \sum_{u=1}^{d_{\text{ffn}}} g_u(\mathbf{x}_i^{\text{mid},l} \mathbf{w}_{\text{in}_u}^l) \mathbf{w}_{\text{out}_u}^l \quad (8)$$

$$= \sum_{u=1}^{d_{\text{ffn}}} n_u^l \mathbf{w}_{\text{out}_u}^l, \quad (9)$$

with $\mathbf{n}^l \in \mathbb{R}^{d_{\text{ffn}}}$ being the vector of neuron activations, and n_u^l the u -th neuron activation value.

The elementwise nonlinearity inside FFNs creates a *privileged basis* (Elhage et al., 2022b), which encourages features to align with basis directions. For instance, given a linear network $f(\mathbf{x}) = \mathbf{x} \mathbf{W}_1 \mathbf{W}_2$, the representations extracted from its first layer, $\mathbf{x} \mathbf{W}_1$, are rotationally invariant, since we can rotate them by an orthogonal matrix \mathbf{O} , giving $\mathbf{x} \mathbf{W}_1 \mathbf{O}$, and invert the rotation having the output of the network untouched, $f(\mathbf{x}) = \mathbf{x} \mathbf{W}_1 \mathbf{O} \mathbf{O}^{-1} \mathbf{W}_2$ (Brown et al., 2023). However, having an elementwise nonlinear function on the output of the first layer breaks the rotational invariance of the representations, making the standard basis dimensions (neurons) more likely to be independently meaningful, and therefore better suitable for interpretability analysis.

2.2 PREDICTION HEAD AND TRANSFORMER DECOMPOSITIONS

The prediction head of a Transformer consists of an unembedding matrix $\mathbf{W}_U \in \mathbb{R}^{d \times |\mathcal{V}|}$, sometimes accompanied by a bias. The last residual stream state gets transformed by this linear map converting the representation into a next-token distribution of logits, which is turned into a probability distribution via the softmax function.

Prediction as a sum of component outputs. The residual stream view shows that every model component interacts with it through addition (Mickus et al., 2022). Thus, the unnormalized scores (logits) are obtained via a linear projection of the summed component outputs. Due to the properties of linear transformations, we can rearrange the traditional forward pass formulation so that each model component contributes directly to the output logits:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{x}_n^L \mathbf{W}_U \\ &= \left(\sum_{l=1}^L \sum_{h=1}^H \text{Attn}^{l,h}(\mathbf{X}_{\leq n}^{l-1}) + \sum_{l=1}^L \text{FFN}^l(\mathbf{x}_n^{\text{mid},l}) + \mathbf{x}_n \right) \mathbf{W}_U \\ &= \sum_{l=1}^L \sum_{h=1}^H \underbrace{\text{Attn}^{l,h}(\mathbf{X}_{\leq n}^{l-1}) \mathbf{W}_U}_{\text{Attention head logits update}} + \sum_{l=1}^L \underbrace{\text{FFN}^l(\mathbf{x}_n^{\text{mid},l}) \mathbf{W}_U}_{\text{FFN logits update}} + \mathbf{x}_n \mathbf{W}_U. \end{aligned} \quad (10)$$

This decomposition plays an important role when localizing components responsible for a prediction (Section 3) since it allows us to measure the direct contribution of every component to the logits of the predicted token (Section 3.2.1).

Prediction as an ensemble of shallow networks forward passes. Residual networks work as ensembles of shallow networks (Veit et al., 2016), where each subnetwork defines a path in the computational graph. Let us consider a two-layer attention-only Transformer, where each attention head is composed just by an OV matrix: $f(\mathbf{x}) = \mathbf{x}^1 + \mathbf{W}_{\text{OV}}^2(\mathbf{x}^1)$, with $\mathbf{x}^1 = \mathbf{x} + \mathbf{W}_{\text{OV}}^1(\mathbf{x})$. We can decompose the forward pass (Figure 3) as

$$f(\mathbf{x}) = \underbrace{\mathbf{x} \mathbf{W}_U}_{\text{Direct path}} + \underbrace{\mathbf{x} \mathbf{W}_{\text{OV}}^1 \mathbf{W}_U}_{\text{Full OV circuits}} + \underbrace{\mathbf{x} \mathbf{W}_{\text{OV}}^1 \mathbf{W}_{\text{OV}}^2 \mathbf{W}_U}_{\text{Full OV circuits}} + \underbrace{\mathbf{x} \mathbf{W}_{\text{OV}}^2 \mathbf{W}_U}_{\text{Full OV circuits}}. \quad (11)$$

↑ Virtual attention heads (V-composition)

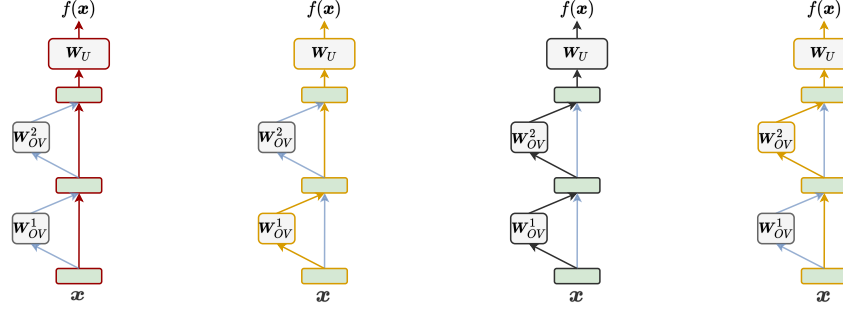


Figure 3: Forward pass decomposition in a simplified Transformer LM. The direct path (red), full OV circuits (yellow) and virtual attention heads (grey) expressed in Equation (11) are highlighted.

The first term in Equation (11), linking the input embedding to the unembedding matrix, is referred to as the *direct path* (first path in Figure 3). The paths traversing a single OV matrix are instead named *full OV circuits* (second and fourth path in Figure 3). Often, full OV circuits are written as $\mathbf{W}_E \mathbf{W}_{OV} \mathbf{W}_U \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, stacking as rows the logits effect of each input embedding through the circuit. Lastly, the path involving both attention heads is referred to as *virtual attention heads* doing *V-composition*, since the sequential writing and reading of the two heads is seen as OV matrices composing together. Elhage et al. (2021a) propose measuring the amount of composition as: $\|\mathbf{W}_{OV}^1 \mathbf{W}_{OV}^2\|_F / \|\mathbf{W}_{OV}^1\|_F \|\mathbf{W}_{OV}^2\|_F$. *Q-composition* and *K-composition*, i.e. compositions of \mathbf{W}_Q and \mathbf{W}_K with the \mathbf{W}_{OV} output of previous layers, can also be found in full Transformer models.

3 BEHAVIOR LOCALIZATION

Understanding the inner workings of language models implies localizing which elements in the forward pass (input elements, representations, and model components) are responsible for a specific prediction.⁶In this section, we present two different types of methods that allow localizing model behavior: *input attribution* (Section 3.1) and *model component attribution* (Section 3.2).

3.1 INPUT ATTRIBUTION

Input attribution methods are commonly used to localize model behavior by estimating the contribution of input elements (in the case of LMs, tokens) in defining model predictions. We refer readers to Madsen et al. (2022) for a broader overview of post-hoc input attribution methods with a focus on classification tasks in NLP.

Gradient-based input attribution. For neural network models like LMs, gradient information is frequently used as a natural metric for attribution purposes (Simonyan et al., 2014; Li et al., 2016; Ding & Koehn, 2021). *Gradient-based attribution* in this context involves a first-order Taylor expansion of a Transformer at a point \mathbf{x} , expressed as $\nabla f(\mathbf{x}) \cdot \mathbf{x} + \mathbf{b}$. The resulting gradient $\nabla f_w(\mathbf{x}) \in \mathbb{R}^{n \times d} = (\text{grad } f_w)(\mathbf{x})$ captures intuitively the *sensitivity* of the model to each element in the input when predicting token w .⁷ While attribution scores are computed for every dimension of input token embeddings, they are generally aggregated at a token level to obtain a more intuitive overview of the influence of individual tokens. This is commonly done by taking the L^p norm of the gradient vector w.r.t the i -th input embedding:

$$A_{f_w(\mathbf{x}) \leftarrow t_i}^{\text{Grad}} = \|\nabla_{\mathbf{x}_i} f_w(\mathbf{x})\|_p. \quad (12)$$

By taking the dot product between the gradient vector and the input embedding $\nabla_{\mathbf{x}_i} f_w(\mathbf{x}) \cdot \mathbf{x}_i$, known as *gradient \times input* method (Denil et al., 2015), this sensitivity can be converted to an importance estimate. However, these approaches are known to exhibit gradient saturation and shattering issues (Shrikumar et al., 2017; Balduzzi et al., 2017). This fact prompted the introduction of methods such as *integrated gradients* (Sundararajan et al., 2017) and SmoothGrad (Smilkov

⁶Commonly referred to as *local explanation* in the interpretability literature (Lipton, 2018)

⁷Vocabulary logits or probability scores are commonly used as differentiation targets (Bastings et al., 2022).

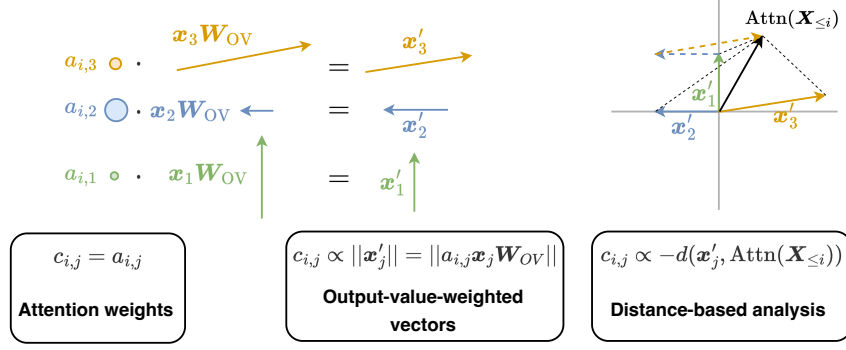


Figure 4: Three approaches to compute inter-token contributions ($c_{i,j}$) towards context mixing in attention heads. Relying only on attention weights overlooks the magnitude of the vectors they operate on. This limitation can be addressed by accounting for the norm of the value-weighted or output-value-weighted vectors (\mathbf{x}'_j). Finally, distance-based analysis estimates the contribution of weighted vectors from their proximity to the attention output.

et al., 2017) to filter noisy gradient information. For example, integrated gradients approximate the integral of gradients along the straight-line path between a baseline input $\tilde{\mathbf{x}}$ and the input \mathbf{x} : $(\mathbf{x}_i - \tilde{\mathbf{x}}_i) \int_0^1 \nabla_{\mathbf{x}_i} f_w(\tilde{\mathbf{x}} + \alpha(\mathbf{x} - \tilde{\mathbf{x}})) d\alpha$, and subsequent adaptations were proposed to accommodate the discreteness of textual inputs (Sanyal & Ren, 2021; Enguehard, 2023). Finally, approaches based on Layer-wise Relevance Propagation (LRP) (Bach et al., 2015) have been widely applied to study Transformer-based LMs (Voita et al., 2021; Chefer et al., 2021; Ali et al., 2022; Achitbat et al., 2024). These methods use custom rules for gradient propagation to decompose component contributions at every layer, ensuring their sum remains constant throughout the network.

Perturbation-based input attribution. Another popular family of approaches estimates input importance by adding noise or ablating input elements and measuring the resulting impact on model predictions (Li et al., 2017). For instance, the input token at position i can be removed, and the resulting probability difference $f_w(\mathbf{x}) - f_w(\mathbf{x}_{-i})$ can be used as an estimate for its importance. If the logit or probability given to w does not change, we conclude that the i -th token has no influence. A multitude of perturbation-based attribution methods exist in the literature, such as those based on *interpretable local surrogate models* such as LIME (Ribeiro et al., 2016), or those derived from *game theory* like SHAP (Shapley, 1953; Lundberg & Lee, 2017). Notably, new perturbation-based approaches were proposed to leverage linguistic structures (Amara et al., 2024; Zhao & Shan, 2024) and Transformer components (Deiseroth et al., 2023; Mohebbi et al., 2023) for attribution purposes. These methods relate directly to causal interventions discussed in Section 3.2.2. We refer readers to Covert et al. (2021) for a unified perspective on perturbation-based input attribution.

Context mixing for input attribution. While raw model internals such as attention weights were generally considered to provide unfaithful explanations of model behavior (Jain & Wallace, 2019; Bastings & Filippova, 2020), recent methods have proposed alternatives to attention weights for measuring intermediate token-wise attributions. Some of these alternatives include the use of the norm of value-weighted vectors (Kobayashi et al., 2020) and output-value-weighted vectors (Kobayashi et al., 2021), or the use of vectors’ distances to estimate contributions (Ferrando et al., 2022b) (Figure 4 provides a visual description). A common strategy among such approaches involves aggregating intermediate per-layer attributions reflecting *context mixing* patterns (Brunner et al., 2020) using techniques such as attention rollout (Abnar & Zuidema, 2020), resulting in input attribution scores (Ferrando et al., 2022b; Modarressi et al., 2022; Mohebbi et al., 2023).⁸ Such context mixing approaches have shown strong faithfulness compared to gradient and perturbation-based methods on classification benchmarks such as ERASER (DeYoung et al., 2020). However, rollout aggregation has recently been criticized due to its simplistic assumptions, and recent research has attempted to fully expand the *linear decomposition* of the model output presented in Equ-

⁸The attention flow method is seldom used due to its computational inefficiency, despite its theoretical guarantees (Ethayarajh & Jurafsky, 2021).

tion (10) (Modarressi et al., 2023; Yang et al., 2023; Oh & Schuler, 2023) as a sum of linear transformations of the input tokens, linearizing the FFN block (Kobayashi et al., 2024).

Contrastive input attribution. An important limitation of input attribution methods for interpreting language models is that attributed output tokens belong to a large vocabulary space, often having semantically equivalent tokens competing for probability mass in next-word prediction (Holtzman et al., 2021). In this context, attribution scores are likely to misrepresent several overlapping factors such as grammatical correctness and semantic appropriateness driving the model prediction. Recent work addresses this issue by proposing a contrastive formulation of such methods, producing counterfactual explanations for why the model predicts token w *instead of* an alternative token o (Yin & Neubig, 2022). As an example, Yin & Neubig (2022) extend the vanilla gradient method of Equation (12) to provide contrastive explanations (ContGrad):

$$A_{f_w \rightarrow o(\mathbf{x}) \leftarrow t_i}^{\text{ContGrad}} = \|\nabla_{\mathbf{x}_i} (f_w(\mathbf{x}) - f_o(\mathbf{x}))\|_p. \quad (13)$$

Limitations of input attribution methods. While input attribution methods are commonly used to debug failure cases and identify biases in models’ predictions (McCoy et al., 2019), popular approaches were shown to be insensitive to variations in the model and data generating process (Adebayo et al., 2018; Sixt et al., 2020), to disagree with each others’ predictions (Atanasova et al., 2020; Crabbé & van der Schaar, 2023; Anonymous, 2024) and to show limited capacity in detecting unseen spurious correlations (Adebayo et al., 2020; 2022). Importantly, popular methods such as SHAP and Integrated Gradients were found provably unreliable at predicting counterfactual model behavior in realistic settings Bilodeau et al. (2024). Apart from theoretical limitations, perturbation-based approaches also suffer from out-of-distribution predictions induced by unrealistic noised or ablated inputs, and from high computational cost of targeted ablations for granular input elements.

Training data attribution. Another dimension of input attribution involves the identification of influential training examples driving specific model predictions at inference time (Koh & Liang, 2017). These approaches are commonly referred to as *training data attribution* (TDA) or *instance attribution* methods and were applied to identify data artifacts (Han et al., 2020; Pezeshkpour et al., 2022) and sources of biases in language models’ predictions (Brunet et al., 2019), with recent approaches proposing to perform TDA via training run simulations (Guu et al., 2023; Liu et al., 2024). While the applicability of established TDA methods was put in question (Akyurek et al., 2022), especially due to their inefficiency, recent work in this area has produced more efficient methods that can be applied to large generative models at scale (Park et al., 2023b; Grosse et al., 2023; Kwon et al., 2024). We refer readers to (Hammoudeh & Lowd, 2022) for further details on TDA methods.

3.2 MODEL COMPONENT IMPORTANCE

Early studies on the importance of Transformers LMs components highlighted a high degree of sparsity in model capabilities. This means, for example, that removing even a significant fraction of the attention heads in a model may not deteriorate its downstream performances (Michel et al., 2019; Voita et al., 2019b). These results motivated a new line of research studying how various components in an LM contribute to its wide array of capabilities.

3.2.1 LOGIT ATTRIBUTION

Let us call $f^c(\mathbf{x})$ the output representation of a model component c (attention head or FFN) at a particular layer for the last token position n . The decomposition presented in Equation (10) allows us to measure the *direct logit attribution*⁹ (DLA, Figure 5) of each model component for the output token $w \in \mathcal{V}$:

$$A_{f_w(\mathbf{x}) \leftarrow c}^{\text{DLA}} = f^c(\mathbf{x}) \mathbf{W}_{U[:,w]}, \quad (14)$$

where $\mathbf{W}_{U[:,w]}$ is the w -th column of \mathbf{W}_U , i.e. the unembedding vector of token w . In practical terms, the DLA for a component c expresses the contribution of c to the logit of the predicted token, using the linearity of the model’s components described in Section 2.2.

⁹Note that the softmax function is shift-invariant, and therefore the logit scores have no absolute scale.

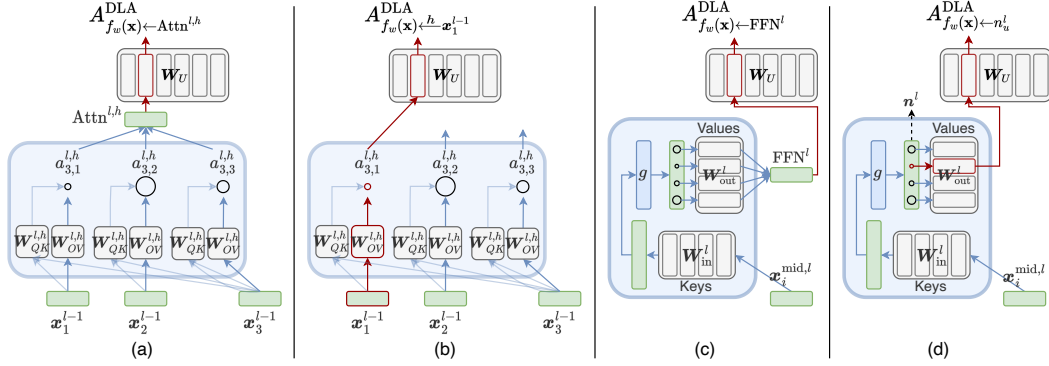


Figure 5: Direct Logit Attributions (DLA) on output token w . (a) DLA of an attention head $\text{Attn}^{l,h}$, (b) DLA of an intermediate representation x_1^{l-1} via an attention head, (c) DLA of an FFN block, and (d) DLA of a single neuron.

Geva et al. (2022b) exploit the fact that the FFN block update is a linear combination of the rows of W_{out} weighted by the neuron activation values (Equation (8)). Thus, it is possible to measure the DLA of each neuron as:

$$A_{f_w(\mathbf{x}) \leftarrow n_u^l}^{\text{DLA}} = n_u^l w_{out_u}^l W_{U[:,w]}, \quad (15)$$

Similarly, Ferrando et al. (2023) makes use of the decomposition of an attention head as a weighted sum of residual stream transformations (Equation (2)) and proposes assessing the DLA of each path involving the attention head:

$$A_{f_w(\mathbf{x}) \leftarrow^h x_j^{l-1}}^{\text{DLA}} = a_{n,j}^{l,h} x_j^{l-1} W_{OV}^{l,h} W_{U[:,w]}. \quad (16)$$

The *Logit difference* (LD) (Wang et al., 2023a) is the difference in logits between two tokens, $f_w(\mathbf{x}) - f_o(\mathbf{x})$. DLA can be extended to measure direct logit difference attribution (DLDA):

$$A_{f_{w-o}(\mathbf{x}) \leftarrow c}^{\text{DLDA}} = f^c(\mathbf{x}) W_{U[:,w]} - f^c(\mathbf{x}) W_{U[:,o]}. \quad (17)$$

including its neuron and head-specific variants of Equation (15) and Equation (16). Similarly to the contrastive attribution framework described in Section 3.1, a positive DLDA value suggests that c promotes token w more than token o .

3.2.2 CAUSAL INTERVENTIONS

We can view the computations of a Transformer-based LM as a causal model (Geiger et al., 2021; McGrath et al., 2023), and use causality tools (Pearl, 2009; Vig et al., 2020) to shed light on the contribution to the prediction of each model component $c \in \mathcal{C}$ across different positions. The causal model can be seen as a directed acyclic graph (DAG), where nodes are model computations and edges are activations. We can intervene in the model by changing some node’s value $f^c(\mathbf{x})$ computed by a model component¹⁰ in the forward pass on *target input* \mathbf{x} , to those from another value \tilde{h} , which is referred to as *activation patching* (Figure 6).¹¹ We can express this intervention using the do-operator (Pearl, 2009) as $f(\mathbf{x}|\text{do}(f^c(\mathbf{x}) = \tilde{h}))$. We then measure how much the prediction changes after patching:

$$A_{f(\mathbf{x}) \leftarrow c}^{\text{Patch}} = \text{diff}(f(\mathbf{x}), f(\mathbf{x}|\text{do}(f^c(\mathbf{x}) = \tilde{h}))). \quad (18)$$

Popular choices for the $\text{diff}(\cdot, \cdot)$ function include KL divergence and logit/probability difference (Zhang & Nanda, 2024). The patched activation (\tilde{h}) can be originated from various sources. A common approach is to create a counterfactual dataset with distribution P_{patch} , where some input signals regarding the task are inverted. This approach leads to two distinct types of ablation:

¹⁰ Alternatively, we can patch residual stream states $f^l(\mathbf{x})$.

¹¹ Also referred to in the literature as Causal Mediation Analysis (Vig et al., 2020), Causal Tracing (Meng et al., 2022), and Interchange Interventions (Geiger et al., 2020; 2021).

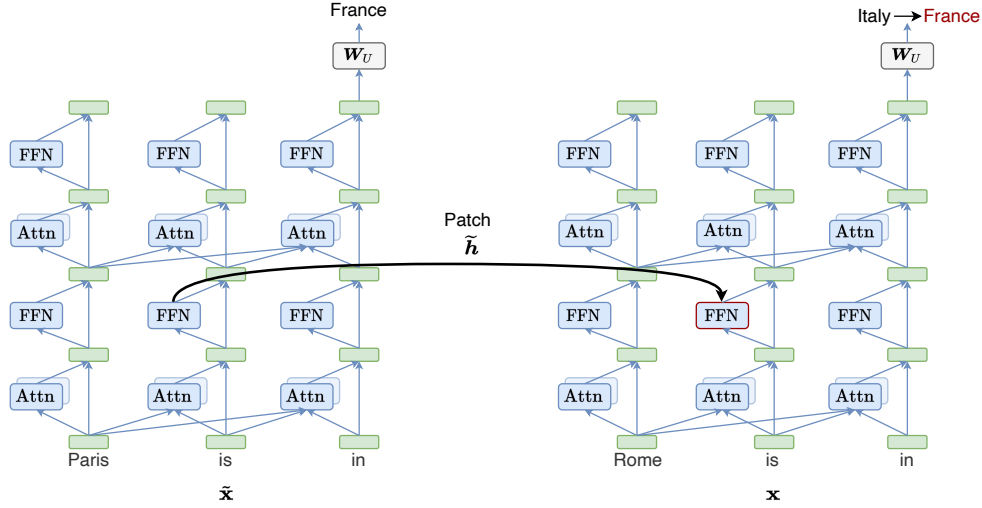


Figure 6: Activation (resample) patching. The FFN output activation from the forward pass with a source input $\tilde{\mathbf{x}}$ (left) is placed in the forward pass with target input \mathbf{x} (right), making the prediction flip from “Italy” to “France”.

- Resample intervention¹², where the patched activation is obtained from a single example of P_{patch} , i.e. $\tilde{\mathbf{h}} = f^c(\tilde{\mathbf{x}})$, $\tilde{\mathbf{x}} \sim P_{\text{patch}}$ (Heimersheim & Janiak, 2023; Hanna et al., 2023; Conmy et al., 2023).
- Mean intervention, where the average of activations of multiple P_{patch} examples is used for patching, i.e. $\tilde{\mathbf{h}} = \mathbb{E}_{\tilde{\mathbf{x}} \sim P_{\text{patch}}} [f^c(\tilde{\mathbf{x}})]$ (Wang et al., 2023a).

Alternatively, other sources of patching activations include:

- Zero intervention, where the activation is substituted by a null vector, i.e. $\tilde{\mathbf{h}} = \mathbf{0}$ (Olsson et al., 2022; Mohebbi et al., 2023).
- Noise intervention, where the new activation is obtained by running the model on a perturbed input, e.g. $\tilde{\mathbf{h}} = f^c(\mathbf{x} + \epsilon)$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ (Meng et al., 2022).

An important factor to consider when designing causal interventions experiments is the *ecological validity* of the setup, since zero and noise ablation could lead the model away from the natural activations distribution and ultimately undermine the validity of components’ analysis (Chan et al., 2022; Zhang & Nanda, 2024).

Following the distinction of Kramár et al. (2024), we note that the activation patching methods presented above adopt a *noising* setup, since the patching is performed during the forward pass with the clean/target input, i.e. $f(\mathbf{x} | \text{do}(f^c(\mathbf{x}) = \tilde{\mathbf{h}}))$ (Wang et al., 2023a; Hanna et al., 2023). Alternatively, the same interventions can be performed in a *denoising* setup, where the patch $\tilde{\mathbf{h}}$ is taken from the clean/target run and applied over the patched run on source/corrupted input, i.e. $f(\tilde{\mathbf{x}} | \text{do}(f^c(\tilde{\mathbf{x}}) = \tilde{\mathbf{h}}))$ (Meng et al., 2022; Lieberum et al., 2023). We refer readers to Heimersheim & Nanda (2024) for a comprehensive overview of metrics, good practices and pitfalls of activation patching.

Other forms of causal interventions use differentiable binary masking on subsets of units or neurons of intermediate representations (De Cao et al., 2020; Csordás et al., 2021; De Cao et al., 2022), or entire attention heads outputs (Voita et al., 2019b; Michel et al., 2019) which can be cast as a form of zero ablation.

Subspace Activation Patching. It is hypothesized that models encode features as linear subspaces of the representation space (Section 4.2). Geiger et al. (2023b) proposed distributed interchange in-

¹²Commonly named *ablation* in the literature. We use the more neutral *intervention* here since activations are not actually ablated, but rather replaced.

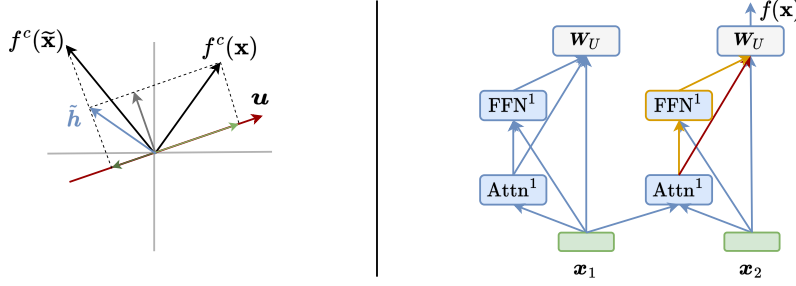


Figure 7: **Left:** Distributed Interchange Interventions (subspace activation patching) on a 1-dimensional subspace (direction) u . **Right:** Single-layer Transformer. Path patching replaces the edges of different paths connecting two nodes (sender and receiver) representing model components. For instance, we can measure the *direct* effect of the attention head on the output $f(\mathbf{x})$ or the *indirect* effect of the attention head on the output $f(\mathbf{x})$ via the FFN.

terventions (DII), which aim to intervene only on these subspaces.¹³ It provides a tool that allows for a fine-grained intervention, rather than relying on patching full representations. Formally, assuming a model component c takes values in \mathbb{R}^d , we seek to find a linear subspace $U \subset \mathbb{R}^d$, where by replacing the orthogonal projection of $f^c(\mathbf{x})$ on U with that of $f^c(\tilde{\mathbf{x}})$ we substitute the feature of interest present in $f^c(\mathbf{x})$ by that in $f^c(\tilde{\mathbf{x}})$. Following the do-operation notation for the intervention process, $f(\mathbf{x}|\text{do}(f^c(\mathbf{x}) = \tilde{h}))$, the patched activation is computed as follows:

$$\tilde{h} = \underbrace{f^c(\mathbf{x}) - f^c(\mathbf{x})U^\top U}_{\text{proj}_{U^\perp} f^c(\mathbf{x})} + f^c(\tilde{\mathbf{x}})U^\top U \quad (19)$$

where $U \in \mathbb{R}^{n \times d}$ is an orthonormal matrix whose rows form a basis for U . If the feature is encoded as a direction (Figure 7 left), i.e. in a 1-dimensional subspace, then the patched activation becomes

$$\tilde{h} = \underbrace{f^c(\mathbf{x}) - \underbrace{f^c(\mathbf{x})u^\top u}_{\text{Target projection}}}_{\text{Target projection subtraction}} + \underbrace{f^c(\tilde{\mathbf{x}})u^\top u}_{\text{Source projection}}. \quad (20)$$

3.2.3 CIRCUITS ANALYSIS

The Mechanistic Interpretability (MI) subfield focuses on reverse-engineering neural networks into human-understandable algorithms (Olah, 2022). Recent studies in MI aim to uncover the existence of *circuits*, which are a subset of model components (subgraphs) interacting together to solve a task (Cammarata et al., 2020). Activation patching, logit attribution, and attention pattern analysis are common techniques for circuit discovery (Wang et al., 2023a; Stolfo et al., 2023a;b; Heimersheim & Janiak, 2023; Geva et al., 2023; Hanna et al., 2023).

Edge and path patching. Activation patching propagates the effect of the intervention throughout the network by recomputing the activations of components after the patched location (Figure 6). The changes in the model output (Equation (18)) allow estimating the *total effect* of the model component on the prediction. However, circuit discovery also requires identifying important interactions between components. For this purpose, *edge patching* exploits the fact that every model component input is the sum of the output of previous components in its residual stream (Section 2.2), and considers edges directly connecting pairs of model components’ nodes (Figure 7 right). *Path patching* generalizes the edge patching approach to multiple edges (Wang et al., 2023a; Goldowsky-Dill et al., 2023), allowing for a more fine-grained analysis. For example, using the forward pass decomposition into shallow networks described in Equation (11), we could visualize the single-layer Transformer of Figure 7 (right) as being composed as

¹³Subspace causal interventions were also used as *causal probes* by Guerner et al. (2023).

$$f(\mathbf{x}) = \text{Attn}(\mathbf{X}_{\leq n}) \mathbf{W}_u + \text{FFN}(\text{Attn}(\mathbf{X}_{\leq n}) + \mathbf{x}_n) \mathbf{W}_u + \mathbf{x}_n \mathbf{W}_u, \quad (21)$$

where each copy of the *sender* node $\text{Attn}^L(\mathbf{X}_{\leq n}^{L-1})$ is relative to a single path. In this example, patching separately each of the sender node copies (Goldowsky-Dill et al., 2023) allows us to estimate *direct* and *indirect* effects (Pearl, 2001; Vig et al., 2020) of $\text{Attn}^L(\mathbf{X}_{\leq n}^{L-1})$ to the output logits $f(\mathbf{x})$. In general, we can apply path patching to any path in the network and measure composition between heads, FFNs, or the effects of these components on the logits.

Limitations of circuit analysis with causal interventions. Circuit analysis based on causal intervention methods presents several shortcomings:

1. it demands significant efforts for designing the input templates for the task to evaluate, along with the counterfactual dataset, i.e. defining P_{patch} .
2. isolating important subgraphs after obtaining component importance estimates requires human inspection and domain knowledge.
3. it has been shown that interventions can produce second-order effects in the behavior of downstream components (Makelov et al., 2024, see Wu et al., 2024d for discussion), in some settings even eliciting compensatory behavior akin to *self-repair* (McGrath et al., 2023; Rushing & Nanda, 2024). This phenomenon can make it difficult to draw conclusions about the role of each component.

Overcoming the limitations. Conmy et al. (2023) propose an Automatic Circuit Discovery (ACDC) algorithm to automate the process of circuit identification (Limitation 2) by iteratively removing edges from the computational graph. However, this process requires a large amount of forward passes (one per patched element), which becomes impractical when studying large models (Lieberum et al., 2023). A valid alternative to patching involves gradient-based methods, which have been extended beyond input attribution to compute the importance of intermediate model components (Leino et al., 2018; Shrikumar et al., 2018; Dhamdhere et al., 2019). For instance, given the token prediction w , to calculate the attribution of an intermediate layer l , denoted as $f^l(\mathbf{x})$, the gradient $\nabla f_w(f^l(\mathbf{x}))$ is computed. Sarti et al. (2023) extend the contrastive gradient attribution formulation of Equation (13) to locate components contributing to the prediction of the correct continuation over the wrong one using a single forward and backward pass. Nanda (2023); Syed et al. (2023) propose Edge Attribution Patching (EAP), consisting of a linear approximation of the pre- and post-patching prediction difference (Equation (18)) to estimate the importance of each edge in the computational graph. The key advantage of this method is that it requires two forward passes and one backward pass to obtain attribution scores of every edge in the graph. Hanna et al. (2024) propose combining EAP with Integrated Gradients (EAP-IG) and show improved faithfulness of the extracted circuits, a method also used by Marks et al. (2024) to identify sparse feature circuits. Further work on Attribution Patching by Kramár et al. (2024) finds two settings leading to false negatives in the linear approximation of activation patching, and proposes AtP*, a more robust method preserving a good computational efficiency. Recently, Ferrando & Voita (2024) propose finding relevant subnetworks, which they name *information flow routes*, using a patch-free context mixing approach, requiring only a single forward pass, avoiding the dependence on counterfactual examples and the risk of self-repair interferences during the analysis.

Causal Abstraction. Another line of research deals with finding interpretable high-level causal abstractions in lower-level neural networks (Geiger et al., 2021; 2022; 2023a). These methods involve a computationally expensive search and assume high-level variables align with groups of units or neurons. To overcome the limitations, Geiger et al. (2023b) propose distributed alignment search (DAS), which performs distributed interchange interventions (DII, Section 3.2.2) on non-basis-aligned subspaces of the low-level representation space found via gradient descent.¹⁴ DAS interventions have been shown to be effective in finding features with causal influence in targeted syntactic evaluation (Arora et al., 2024), and in isolating the causal effect of individual attributes of entities (Huang et al., 2024a). Recently, learned edits on subspaces of intermediate representations during the forward pass have been proposed as an efficient and effective alternative to weight-based

¹⁴Alternatively, Lepori et al. (2023) proposes employing circuit discovery approaches for this purpose.

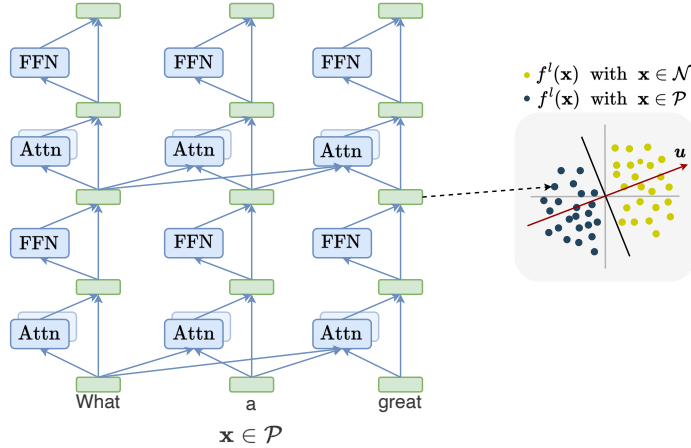


Figure 8: A binary probe trained to predict the input sentiment with positive \mathcal{P} and negative \mathcal{N} sentences. Binary linear classifier probes work within a 1-dimensional subspace (direction u) in the representation space.

Parameter-efficient fine-tuning (PEFT) approaches (Wu et al., 2024b). A DAS variant named Boundless DAS has been used to search for interpretable causal structure in large language models (Wu et al., 2023b). In this context, Causal Proxy Models (CPMs) were proposed as interpretable proxies trained to mimic the predictions of lower-level models and simulate their counterfactual behavior after targeted interventions (Wu et al., 2023a).

4 INFORMATION DECODING

Fully understanding a model prediction entails localizing the relevant parts of the model, but also comprehending what information is being extracted and processed by each of these components. For example, if the grammatical gender of nouns is assumed to be relevant for the task of coreference resolution in a given language, information decoding methods could look at whether and how a model performing this task encodes noun gender. A natural way to approach decoding the information in the network is in terms of the *features* that are represented in it. While there is no universally agreed-upon definition of a feature, it is typically described as a human-interpretable property of the input¹⁵, which can be also referred to as a *concept* (Kim et al., 2018).

4.1 PROBING

Probes, introduced concurrently in NLP by Köhn (2015); Gupta et al. (2015) and in computer vision by Alain & Bengio (2016) serve as tools to analyze the internal representations of neural networks. Generally, they take the form of supervised models trained to predict input properties from the representations, aiming to assess how much information about the property is encoded in them. Formally, the probing classifier $p : f^l(\mathbf{x}) \mapsto z$ maps intermediate representations to some input features (labels) z , which can be, for instance, a part-of-speech tag (Belinkov et al., 2017), or semantic and syntactic information (Peters et al., 2018). For example, for a binary probe seeking to decode the amount of input sentiment information within an intermediate representation (Figure 8) we build two sets: $\{f^l(\mathbf{x}) : \mathbf{x} \in \mathcal{P}\}$ and $\{f^l(\mathbf{x}) : \mathbf{x} \in \mathcal{N}\}$, with the representations obtained when providing positive and negative sentiment sentences respectively. After training the classifier we evaluate the accuracy results on a held-out set.

Although performance on the probing task is interpreted as evidence for the amount of information encoded in the representations, there exists a tension between the ability of the probe to evaluate the information encoded and the probe learning the task itself (Belinkov, 2022). Several works propose using baselines to contextualize the performance of a probe. Hewitt & Liang (2019) use

¹⁵Although we have evidence that models learn human-interpretable features even in instances that exceed human performance (McGrath et al., 2022), Olah (2022) argues that the definition of feature should include properties that are not human-interpretable.

control tasks by randomizing the probing dataset, while Pimentel et al. (2020) propose measuring the information gain after applying *control functions* on the internal representations. Voita & Titov (2020) suggest evaluating the quality of the probe together with the “amount of effort” required to achieve the quality. This is done by measuring the minimum description length of the code required to transmit labels z given representations $f^l(\mathbf{x})$. We refer the reader to Belinkov & Glass (2019); Belinkov (2022) for a larger coverage of probing methods.

Probing techniques have been largely applied to analyze Transformers in NLP. Although probes are still being used to study decoder-only models (CH-Wang et al., 2023; Zou et al., 2023; Burns et al., 2023; MacDiarmid et al., 2024), a significant portion of the research in this area has focused on BERT (Devlin et al., 2019) and its variants, leading to several BERTology analyses (Rogers et al., 2021). Probing has provided evidence of the existence of syntactic information within BERT representations (Tenney et al., 2019b; Lin et al., 2019; Liu et al., 2019), from which even full parse trees can be recovered with good precision (Hewitt & Manning, 2019). Additionally, some studies have analyzed where syntactic information is stored across the residual stream suggesting a hierarchical encoding of language information, with part-of-speech, constituents, and dependencies being represented earlier in the network than semantic roles and coreferents, matching traditional handcrafted NLP pipelines (Tenney et al., 2019a). Rogers et al. (2021) summarizes results on BERT in detail. Importantly, highly accurate probes indicate a correlation between input representations and labels, but do not provide evidence that the model is using the encoded information for its predictions (Hupkes et al., 2018; Belinkov & Glass, 2019; Elazar et al., 2021).

4.2 LINEAR REPRESENTATION HYPOTHESIS AND SPARSE AUTOENCODERS

Linear Representation Hypothesis. The *linear representation hypothesis* states that features are encoded as linear subspaces of the representation space (see Park et al. (2023a) for a formal discussion). Mikolov et al. (2013) were the first to show that Word2Vec word embeddings capture linear syntactic/semantic word relationships. For example, adding the difference between word representations of “Spain” and “Madrid”, $f(\text{“Spain”}) - f(\text{“Madrid”})$, to the “France” representation, $f(\text{“France”})$, would result in a vector close to $f(\text{“Paris”})$. This presumes that the vector $f(\text{“Spain”}) - f(\text{“Madrid”})$ can be considered as the direction of the abstract *capital_of* feature. Instances of interpretable neurons (Radford et al., 2017; Voita et al., 2023; Bau et al., 2020), i.e. neurons that fire consistently for specific input features (either monosemantic or polysemantic), also exemplify features represented as directions in the neuron space. Recent work suggests the linearity of concepts in representation space is largely driven by the next-word-prediction training objective and inductive biases in gradient descent optimization (Jiang et al., 2024).

Erasing Features with Linear Interventions Feature directions can be found in LMs using linear classifiers (*linear probes*, Section 4.1). These models learn a hyperplane that separates representations associated with a particular feature from the rest. The normal vector to that hyperplane, the probe direction $\mathbf{u} \in \mathbb{R}^d$, can be considered the direction representing the underlying feature (Figure 8). For instance, the sensitivity of model predictions to a feature can be computed as the directional derivative of the model in the direction \mathbf{u} , $\nabla f(f^l(\mathbf{x})) \cdot \mathbf{u}$, treating the model as a function of the intermediate activation (Kim et al., 2018). This linear feature representation was exploited by Ravfogel et al. (2020; 2022); Belrose et al. (2023b) to erase concepts, preventing linear classifiers from detecting them in the representation space. Linear concept erasure was shown to mitigate bias (Ravfogel et al., 2020) or induce a large increase in perplexity after removing part-of-speech information (Belrose et al., 2023b). In presence of class labels, linear erasure models can be adapted to ensure the removal of all linear information regarding class identity (Singh et al., 2024c; Belrose, 2023). Finally, Elazar et al. (2021) exploits linear erasure to address the correlational nature of probing classifier, validating the influence of probed properties on model predictions.

Steering Generation with Linear Interventions As mentioned in Section 4.1, a fundamental problem of probing lies in its correlational, rather than causal, nature. Recent work (Nanda et al., 2023b; Zou et al., 2023) shows the effectiveness of linear interventions on language models using directions identified by a probe. For instance, adding negative multiples of the sentiment direction (\mathbf{u}) to the residual stream, i.e. $\mathbf{x}^{l'} \leftarrow \mathbf{x}^l - \alpha \mathbf{u}$, is sufficient to generate a text matching the opposite sentiment label (Tigges et al., 2023). This simple procedure is named *activation addition* (Turner et al., 2023). Other unsupervised methods for computing features directions include Principal Com-

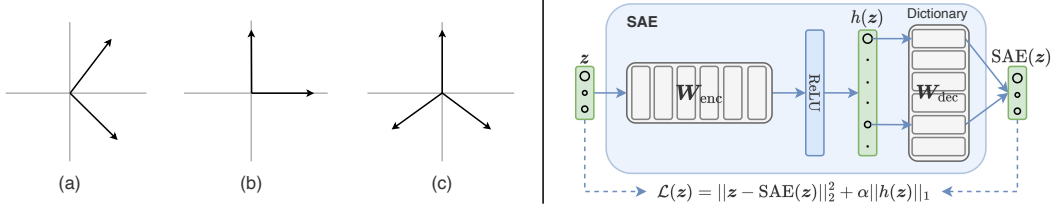


Figure 9: **Left:** Feature directions in a 2-dimensional space. (a) features as directions not aligned with the standard basis, we observe polysemanticity. (b) features aligned with the standard basis, monosemanticity. (c) more features than dimensions (superposition), hence features can’t align with the standard basis and polysemanticity is inevitable. **Right:** Sparse autoencoder (SAE) trained to reconstruct a model’s internal representations z . Interpretable SAE features are found in rows of W_{dec} . Biases are omitted for the sake of clarity.

ponent Analysis (Tigges et al., 2023), K-Means (Zou et al., 2023), or difference-in-means (Marks & Tegmark, 2023). For instance, Arditi et al. (2024) use the difference-in-means vector between residual streams on harmful and harmless instructions to find a “refusal direction” in LMs with safety fine-tuning (Bai et al., 2022). Projecting out this direction from every model component output, i.e. $f^{c'} \leftarrow f^c - f^c u^\top u$, leads to bypass refusal. Recent studies set distributed alignment search (Section 3.2.3) as the best performing method for causal intervention across mathematical reasoning and linguistic plausibility benchmarks (Tigges et al., 2023; Arora et al., 2024; Huang et al., 2024a), and leveraged it for efficient inference-time interventions aimed at improving task-specific model performance (Wu et al., 2024b). Finally, the MiMic framework (Singh et al., 2024c) was recently proposed to craft optimal steering vectors, exploiting insights from linear erasure methods and class labels from the data distribution. We note that the effectiveness of steering approaches involving linear interventions was recently observed to extend to non-Transformer LMs (Paulo et al., 2024).

Polysemanticity and Superposition. A representation produced by a model layer is a vector that lies in a d -dimensional space. Neurons are the special subset of representation units right after an element-wise non-linearity (Section 2.1.3). Although previous work has identified neurons in models corresponding to interpretable features, in most cases they respond to apparently unrelated inputs, i.e. they are *polysemantic*. Two main reasons can explain polysemanticity. Firstly, features can be represented as linear combinations of the standard basis vectors of the neuron space (Figure 9 left (a)), not corresponding to the basis elements themselves. Therefore, each feature is represented across many individual neurons, which is known as *distributed representations* (Smolensky, 1986; Olah, 2023). Secondly, given the extensive capabilities and long-tail knowledge demonstrated by large language models, it has been hypothesized that models could encode more features than they have dimensions, a phenomenon called *superposition* (Figure 9 left (c)) (Arora et al., 2018; Olah et al., 2020b). Elhage et al. (2022b) showed on toy models trained on synthetic datasets that superposition happens when forcing sparsity on features, i.e. making them less frequent on the training data. Recently, Gurnee et al. (2023) have provided evidence of superposition in the early layers of a Transformer language model, using sparse linear probes.

Sparse Autoencoders (SAEs). A possible strategy to disentangle features in superposition involves finding an overcomplete feature basis via dictionary learning (Olshausen & Field, 1997). Autoencoders with sparsity regularization, also known as *sparse autoencoders* (SAEs), can be used for dictionary learning by optimizing them to reconstruct internal representations $z \in \mathbb{R}^d$ of a neural network exhibiting superposition while simultaneously promoting feature sparsity. That is, we obtain a reconstruction $z = SAE(z) + \epsilon$, where ϵ is the *SAE error term*. Sharkey et al. (2022); Bricken et al. (2023); Cunningham et al. (2023) propose training SAEs (see Figure 9 right) of the form

$$SAE(z) = \text{ReLU}((z - b_{dec})W_{enc} + b_{enc}) W_{dec} + b_{dec} \quad (22)$$

\downarrow SAE feature activations $h(z)$
 \downarrow Dictionary SAE features

on language models’ representations with a loss defined as

$$\mathcal{L}(z) = \underbrace{\|z - SAE(z)\|_2^2}_{\text{Reconstruction loss term}} + \underbrace{\alpha \|h(z)\|_1}_{\text{Sparsity loss term}}. \quad (23)$$

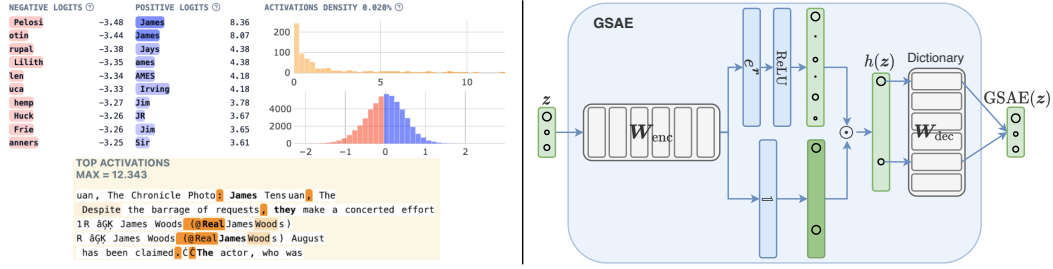


Figure 10: **Left:** SAE feature visualization on Neuronpedia (Lin & Bloom, 2024). It shows the promoted/suppressed tokens, feature density, logits distribution, and maximally activating examples of a *name mover* feature found in GPT-2 Small (Kissane et al., 2024b). **Right:** Gated Sparse Autoencoder with encoder weight sharing. Biases are omitted for the sake of clarity.

By inducing sparsity on the latent representation of SAE feature activations $h(z) = \text{ReLU}(zW_{enc} + b) \in \mathbb{R}^m$ and setting $m > d$, we can approximate z as a sparse linear combination of the rows of the learned $W_{dec} \in \mathbb{R}^{m \times d}$ dictionary, from which we can extract interpretable and monosemantic SAE features.¹⁶ Since the output weights of each SAE feature interact linearly with the residual stream, we can measure their direct effect on the logits (Section 3.2.1) and their composition with later layers’ components (Section 2.2) (He et al., 2024). An initial assessment of reconstruction errors (ϵ) in SAEs trained on LM activations highlighted their systematic nature, driving a shift in next token prediction probabilities much higher than random noise (Gurnee, 2024). Marks et al. (2024) also found these errors account for 1–15% of z variance. While this finding might undermine the faithfulness of component analyses relying on SAE features, Marks et al. (2024) proposes an adaptation of the causal model framework outlined in Section 3.2.2 aiming to incorporate SAE features and errors as nodes of the computational graph. Using edge attribution patching (Section 3.2.3), they recover sparse feature circuits providing more intuitive overviews of features driving model predictions.

SAEs Evaluation. The goal of SAEs is to learn sparse reconstructions of representations. To assess the quality of a trained SAE in achieving this it is common to compute the *Pareto frontier* of two metrics on an evaluation set (Bricken et al., 2023). These metrics are:

- The L_0 norm of the feature activations vector $h(z)$, which measures how many features are “alive” given an input. This metric is averaged across the evaluation set, $\mathbb{E}_{z \sim \mathcal{D}} \|h(z)\|_0$.
- The *loss recovered*, which reflects the percentage of the original cross-entropy loss of the LM across a dataset when substituting the original representations with the SAE reconstructions.

A summary statistic proposed by Bricken et al. (2023) is the *feature density histogram*. *Feature density* is the proportion of tokens in a dataset where a SAE feature has a non-zero value. By looking at the distribution of feature densities we can distinguish if the SAE learnt features that are too dense (activate too often) or too sparse (activate too rarely). Finally, the degree of *interpretability of sparse features* can be estimated based on their *direct logit attribution* and *maximally activating examples* (see Figure 10 left, we introduce these concepts in Section 4.3). This process can be done manually or automated, using a LLM to produce natural language explanations of SAE features.

Gated SAEs (GSAEs). The sparsity penalty used in SAE training promotes smaller feature activations, biasing the reconstruction process towards smaller norms. This phenomenon is known as *shrinkage* (Tibshirani, 1996; Wright & Sharkey, 2024). Rajamanoharan et al. (2024) address this issue by proposing Gated Sparse Autoencoders (GSAEs) and a complementary loss function. GSAE is inspired by Gated Linear Units (Dauphin et al., 2017; Shazeer, 2020), which employ a gated ReLU encoder to decouple feature magnitude estimation from feature detection (Figure 10 right):

¹⁶We present in Appendix D some SAEs’ implementation details currently debated.

$$\text{GSAE}(z) = \underbrace{\mathbb{1}[(z - \mathbf{b}_{\text{dec}})\mathbf{W}_{\text{gate}} + \mathbf{b}_{\text{gate}} > 0]}_{h(z)} \odot \underbrace{\text{ReLU}((z - \mathbf{b}_{\text{dec}})\mathbf{W}_{\text{mag}} + \mathbf{b}_{\text{mag}})}_{\text{GSAE feature activations' magnitude}} \mathbf{W}_{\text{dec}} + \mathbf{b}_{\text{dec}}, \quad (24)$$

where $\mathbb{1}$ is the step function. The features' gate and activation magnitudes are computed by sharing weight matrices, $\mathbf{W}_{\text{mag}[i,j]} = \mathbf{W}_{\text{gate}[i,j]} e^{\mathbf{r}[j]}$, being $\mathbf{r} \in \mathbb{R}^m$ a learned rescaling vector, thus \mathbf{W}_{gate} can be considered the encoder matrix \mathbf{W}_{enc} (Figure 10). Rajamanoharan et al. (2024) show GSAE is a Pareto improvement over the standard SAE architecture on a range of models, scaling GSAEs up to Gemma 7B (Gemma Team et al., 2024).

4.3 DECODING IN VOCABULARY SPACE

The model engages with the vocabulary in two primary ways: firstly, through a set of input tokens facilitated by the embedding matrix \mathbf{W}_E , and secondly, by interacting with the output space via the unembedding matrix \mathbf{W}_U . Hence, and due to its interpretable nature, a sensible way to approach decoding the information within models' representations is via vocabulary tokens.

Decoding intermediate representations. The *logit lens* (nostalgebraist, 2020) proposes projecting intermediate residual stream states \mathbf{x}^l by \mathbf{W}_U . The logit lens can also be interpreted as the prediction the model would do if skipping all later layers, and can be used to analyze how the model refines the prediction throughout the forward pass (Jastrzębski et al., 2018). This technique has proven effective in analyzing encoder representations in encoder-decoder models Langedijk et al. (2023). However, the logit lens can fail to elicit plausible predictions in some particular models Belrose et al. (2023a). This phenomenon have inspired researchers to train *translators*, which are functions applied to the intermediate representations prior to the unembedding projection. Din et al. (2023) suggest using linear mappings, while Belrose et al. (2023a) propose affine transformations (*tuned lens*). Translators have also been trained on the outputs of attention heads, resulting in the *attention lens* (Sakarvadia et al., 2023). More generally, we can also think of \mathbf{W}_U as the weights learned by a probe whose classes are the subwords in the vocabulary (Section 4.1), and inspect at any point in the network the amount of information encoded about any subword.

Patchscopes. *Patchscopes* (Ghandeharioun et al., 2024) is a framework that generalizes patching to decode information from intermediate representations.¹⁷ Recall from Section 3.2.2 that patching an activation into a forward pass $f(\mathbf{x}|\text{do}(f^c(\mathbf{x}) = \tilde{\mathbf{h}}))$ serves to evaluate the output change with respect to the original clean run $f(\mathbf{x})$. Patchscope defines a function acting on the patched representation $m(\tilde{\mathbf{h}})$, a target model f^* for the patched run, which can differ from the original f , a target prompt \mathbf{x}^* , and a target model component c^* that can be at a different position and layer. It then evaluates $f^*(\mathbf{x}^*|\text{do}(f^c(\mathbf{x})^* = m(\tilde{\mathbf{h}})))$, either by inspecting the output logits, probabilities, or generating from it a natural language explanation. The election of f^* , m , \mathbf{x}^* , and c^* defines the type of information to extract from $\tilde{\mathbf{h}}$ independently of the original context, allowing a higher expressivity. For instance, the *future lens* (Pal et al., 2023), used to decode future tokens from intermediate representations can be considered as a patchscope where $f^* = f$, $c^* = c$ and \mathbf{x}^* is a learned prompt.

Decoding model weights. As seen in previous sections, \mathbf{W}_{OV}^h , \mathbf{W}_{out} , \mathbf{W}_{in} and \mathbf{W}_{QK} interact linearly with the residual streams. Dar et al. (2023) suggest analyzing matrix weights in vocabulary space by projecting them by \mathbf{W}_U , and find that some weight matrices interact with tokens with related semantic meanings. Millidge & Black (2022) propose to factorize these matrices via the singular value decomposition (SVD). In the "thin" SVD, a matrix is factorized as $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, with $\mathbf{U} \in \mathbb{R}^{d \times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$, $\mathbf{V}^\top \in \mathbb{R}^{r \times d}$, and $r = \text{rk}(\mathbf{W})$, the rank of \mathbf{W} . The largest right singular vectors (rows of \mathbf{V}^\top)¹⁸ represent the directions along which a linear transformation stretches the most. Then, multiplying \mathbf{z} by \mathbf{W} (Figure 11 left) can be expressed as

$$\mathbf{z}\mathbf{W} = (\mathbf{z}\mathbf{U}\mathbf{\Sigma})\mathbf{V}^\top = \sum_{i=1}^r (\mathbf{z}\mathbf{u}_i\sigma_i)\mathbf{v}_i^\top. \quad (25)$$

¹⁷A concurrent similar approach is presented by Chen et al. (2024c).

¹⁸Note that we left multiply by \mathbf{W} .

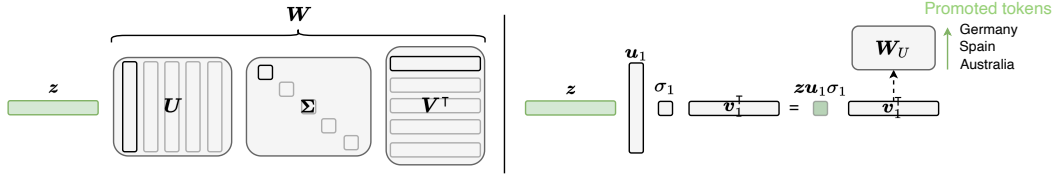


Figure 11: **Left:** Multiplication of an internal representation z by the SVD decomposition of a matrix W . **Right:** The top right singular vector v_1^T represents the direction along which the transformation stretches the most, revealing the tokens the matrix primary interacts with when projecting onto the vocabulary space. The input representation z and the associated left singular vector u_1 act as a query and a key respectively, being v_1^T the value associated with the key.

where $u_i \in \mathbb{R}^{d \times 1}$ can be seen as a key that is compared to the query (z) via dot product, weighting the right singular vector v_i^T (McDougall, 2023; Molina, 2023), similar to Equation (8). By projecting the top right singular vectors onto the vocabulary space via the unembedding matrix ($v_i^T W_U$) we reveal the tokens the matrix primarily interacts with (Figure 11 right). We can instead use the SVD to find a low-rank approximation $\widehat{W}(k) = \sum_{i=1}^k (u_i \sigma_i) v_i^T$, where $\text{rk}(\widehat{W}(k)) = k < r$, and study the model predictions by substituting the original matrix by $\widehat{W}(k)$ (Sharma et al., 2024b). Katz et al. (2024) propose extending the projection of weight matrices (Dar et al., 2023) to the backward pass. Specifically, the *backward lens* projects the gradient matrices of the FFNs to study how new information is stored in their weights.

Logit spectroscopy. Cancedda (2024) proposes an extension of the logit lens, the *logit spectroscopy*, which allows a fine-grained decoding of the information of internal representations via the unembedding matrix (W_u). Logit spectroscopy considers splitting the right singular matrix of W_u into N bands: $\{V_{u,1}^T, \dots, V_{u,N}^T\}$, where $V_{u,1}^T$ and $V_{u,N}^T$ each contain a set of singular vectors, the former associated with the largest singular values and the latter with the lowest. If we consider the concatenation of matrices associated with different bands, e.g. from the j -th to the k -th band, we form a matrix $V_{u,j:k}^T$ whose rows span a linear subspace of the vocabulary space. We can use the operator $\Phi_{u,j:k} = V_{u,j:k}^T V_{u,j:k}^T$ to evaluate the orthogonal projection $z \Phi_{u,j:k}$ of representations z onto different subspaces. Alternatively, we can suppress the projection from the representation, i.e. $z' \leftarrow z - z \Phi_{u,j:k}$, leaving its orthogonal component with respect to the subspace. Similarly, bands of singular vectors of the embedding matrix can be considered in the analysis.

Maximally-activating inputs. The features encoded in model neurons or representation units have been largely studied by considering the inputs that maximally activate them (Zhou et al., 2015; Zeiler & Fergus, 2014). In image models this can be done either by generating synthesized inputs (Nguyen et al., 2016), e.g. via gradient descent (Simonyan et al., 2014), or by selecting examples from an existing dataset. The latter approach has been used in language models to explain the features that units (Dalvi et al., 2019) and neurons (Nanda, 2022b) respond to. However, Bolukbasi et al. (2021) warn that just relying on maximum activating dataset examples can result in “interpretability illusions”, as different activation ranges may lead to varying interpretations. Maximally-activating inputs can produce out-of-distribution behaviors, and were recently employed to craft *jailbreak attacks* aimed at eliciting unacceptable model predictions (Chowdhury et al., 2024), for example by crafting maximally-inappropriate inputs for red-teaming purposes (Wichers et al., 2024).

Natural language explanations from LMs. Modern LMs can be prompted to provide plausible-sounding justifications for their own or other LMs’ predictions. This can be seen as an edge case of information decoding in which the predictor itself is used as a zero-shot explainer. A notable example is the work by Bills et al. (2023) where GPT-4 is prompted to describe shared features in sets of examples producing high activations for specific neurons across GPT-2 XL. Subsequent work by Huang et al. (2023) shows that neurons identified by Bills et al. (2023) do not have a causal influence over the concepts highlighted in the generated explanation, underscoring a lack of faithfulness in such approach. Additional investigations in the consistency between input attribution and self-explanations in language models highlighted the tendency of LMs to produce explanations that are very plausible according to human intuition, but unfaithful to model inner workings (Atanasova

et al., 2023; Parcalabescu & Frank, 2023; Turpin et al., 2023; Lanham et al., 2023; Madsen et al., 2024; Agarwal et al., 2024).

5 DISCOVERED INNER BEHAVIORS

The techniques presented in Sections 3 and 4 have equipped us with essential tools to understand the behavior of language models. In the following sections, we provide an overview of the internal mechanisms that have been discovered within Transformer LMs.

5.1 ATTENTION BLOCK

As seen in Section 2.1.2, each attention head consists of a QK (query-key) circuit and an OV (output-value) circuit. The QK circuit computes the attention weights, determining the positions that need to be attended, while the OV circuit moves (and transforms) the information from the attended position into the current residual stream. A substantial body of research has been dedicated to analyzing attention weights patterns formed by QK circuits (Clark et al., 2019; Kovaleva et al., 2019; Voita et al., 2019b), fueling a debate on whether these weights serve as explanations (Bibal et al., 2022). However, our understanding of the specific features encoded in the subspaces employed by circuit operations is still limited. Here, we categorize known behavior of attention heads in two groups: those having intelligible attention patterns, and those with meaningful QK and OV circuits.

5.1.1 ATTENTION HEADS WITH INTERPRETABLE ATTENTION WEIGHTS PATTERNS

Positional heads. Clark et al. (2019) showed some BERT heads attend mostly to specific positions relative to the token processed. Specifically, attention heads that attend to the token itself, to the previous token, or to the next position. A similar pattern is also observed in encoders of neural machine translation models (Voita et al., 2019b; Raganato & Tiedemann, 2018). **Previous token heads** are an essential part of induction heads, and have been shown necessary for circuits in GPT2-Small (Wang et al., 2023a). Their main role has been associated with copying previous token information to the following residual stream, such as concatenating two-tokens names (Nanda et al., 2023c). Ferrando & Voita (2024) show previous token heads are important across several textual domains.

Subword joiner heads. First discovered in machine translation encoders Correia et al. (2019), subword joiner heads have been observed as well in large language models (Ferrando & Voita, 2024). These heads attend exclusively to previous tokens that are subwords belonging to the same word as the currently processed token.

Syntactic heads. Some attention heads attend to tokens having syntactic roles with respect to the processed token significantly more than a random baseline (Clark et al., 2019; Htut et al., 2019). Particularly, certain heads specialize in given dependency relation types such as obj, nsubj, advmod, and amod. Chen et al. (2024a) show these heads appear suddenly during the training process of masked language models playing a crucial role in the subsequent development of linguistic abilities.

Duplicate token heads. Duplicate token heads attend to previous occurrences of the same token in the context of the current token. Wang et al. (2023a) hypothesize that, in the IOI task (Section 5.4), these heads copy the position of the previous occurrence to the current position.

5.1.2 ATTENTION HEADS WITH INTERPRETABLE QK AND OV CIRCUITS

Copying heads. Several attention heads in Transformer LMs have OV matrices that exhibit copying behavior. Elhage et al. (2021a) propose using the number of positive real eigenvalues of the full OV circuit matrix $W_E W_{OV} W_U$ as a summary statistic for detecting copying heads. Positive eigenvalues mean that there exists a linear combination of tokens contributing to an increase in the linear combination of logits of the same tokens.

Induction heads. An induction mechanism (Figure 12 left) that allows language models to complete patterns was discovered first by Elhage et al. (2021a) and further studied by Olsson et al.

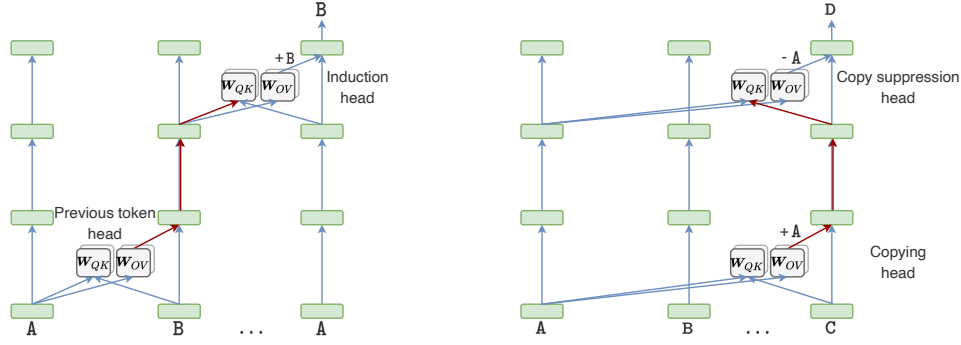


Figure 12: **Left: Induction mechanism.** An early previous token head writes information of A into B’s residual stream via W_{OV} . This information (shown in red) gets read by the W_{QK} matrix of a downstream induction head (K-composition), which serves it to attend to B and copy its information, increasing the likelihood of B for the next token prediction. **Right: Copy suppression mechanism.** The copy suppression head detects that a token in the context (A) is being confidently predicted at the current residual stream, for instance, thanks to a previous copying head (shown in red). The copy suppression head attends to it and suppresses its prediction, improving model calibration. Other components are hidden for the sake of clarity.

(2022).¹⁹ This mechanism involves two heads in different layers composing together. Specifically, a previous token head (PTH) and an induction head. The induction mechanism learns to increase the likelihood of token B given the sequence A B ... A, irrespective of what A and B are. To do so, a PTH in an early layer copies information from the first instance of token A to the residual stream of B, specifically by writing in the subspace the QK circuit of the induction head reads from (K-composition). This makes the induction head at the last position to attend to token B, and subsequently, its copying OV circuit increases the logit score of B. Olsson et al. (2022) demonstrate that the OV and QK circuits of the induction head can perform fuzzy versions of copying and prefix matching, giving rise to generating patterns of the kind $A^* B^* \dots A \rightarrow B$, where A and A^* , and B and B^* are semantically related (e.g. the same words in different languages). Overall, induction heads have been shown to appear broadly in Transformer LMs (Nanda, 2022a), with those operating at an n-gram level being identified as important drivers of in-context learning (Akyürek et al., 2024). Recent work showed that these heads display both complementary and redundant behaviors, likely shaped by competitive dynamics during optimization (see Section 5.4) (Singh et al., 2024a). Relatedly, redundancy was also observed in the connections between early-layer PTHs and subsequent induction heads. Finally, the emergence rate of induction heads is impacted by the diversity of in-context tokens, with higher diversity in attended and copied tokens delaying the formation of the two respective sub-mechanisms (Singh et al., 2024a).

Copy suppression heads. Copy suppression heads, discovered in GPT2-Small (McDougall et al., 2023) reduce the logit score of the token they attend to, only if it appears in the context and the current residual stream is confidently predicting it (Figure 12 right). This mechanism was shown to improve overall model calibration by avoiding naive copying in many contexts (e.g. copying “love” in “All’s fair in love and ____”). The OV circuit of a copy suppression head can copy-suppress almost all of the tokens in the model’s vocabulary when attended to. This behavior is confirmed by analyzing the “effective QK circuit” of GPT2-Small. The key input is the FFN¹ output of every token, and the query input the unembedding of any token, $W_U W_{QK} \text{FFN}^1(W_E)$, and shows the diagonal elements rank higher. Copy suppression is also linked to the self-repair mechanism since ablating an essential component deactivates the suppression behavior, compensating for the ablation.

Successor heads. Given an input token belonging to an element in an ordinal sequence (e.g. “one”, “Monday”, or “January”), the ‘effective OV circuit’: $\text{FFN}^1(W_E) W_{OV} W_U$ of the successor heads increases the logits of tokens corresponding to the next elements in the sequence (e.g. “two”, “Tuesday”, “February”). Specifically, Gould et al. (2024) show the output of the first FFN block represents a common ‘numerical structure’ on which the successor head acts. Gould et al. (2024) find these

¹⁹We follow the mechanistic formulation by Elhage et al. (2021a). See (Variengien, 2023) for a discussion.

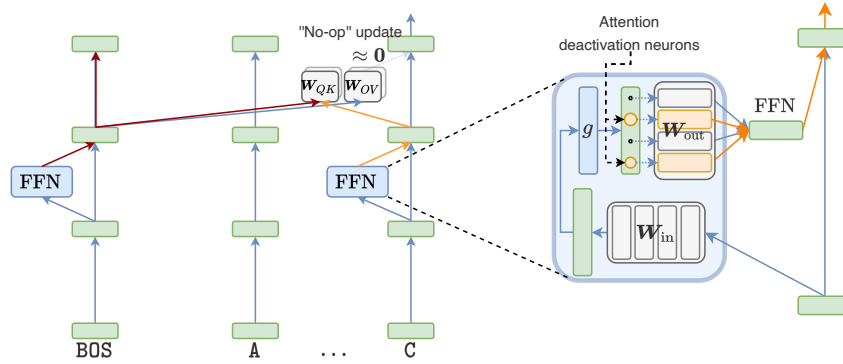


Figure 13: **Attention sink mechanism.** An attention head at a token C attends to BOS token. Its OV circuit squeezes the BOS residual stream representation resulting in a negligible update, leaving the residual stream of C unchanged. Cancedda (2024) suggests that early FFNs in Llama 2 write into a “dark subspace” (in red) in the BOS residual stream that allows later heads to exploit this behavior. Gurnee et al. (2024) find specific neurons in previous layer FFNs of GPT-2 that control the extent to which the query attends to BOS (in orange).

heads in Pythia (Biderman et al., 2023), GPT2 (Radford et al., 2019) and Llama 2 (Touvron et al., 2023) models.

5.1.3 OTHER NOTEWORTHY ATTENTION PROPERTIES

Domain specialization. The attention heads previously described serve specific functions in aiding the model to predict the next token. However, the degree of specialization of components across different domains and tasks remains unclear. Ferrando & Voita (2024); Chughtai et al. (2024); Lv et al. (2024) identify some **specialized heads** that contribute only within specific input domains, such as non-English contexts, coding sequences, or specific topics. An analysis of the top singular vectors of their OV matrices (Section 4.3) reveal these heads mainly promote tokens related to the semantics of the input they participate in.

Attention sinks. Early investigations into BERT (Kovaleva et al., 2019) revealed most attention heads exhibit “vertical” attention patterns, mainly focusing on special (CLS, SEP) and punctuation tokens. Clark et al. (2019) hypothesized a head may attend to special tokens when its specialized function is not applicable (no-op hypothesis). Kobayashi et al. (2020) showed the norm of the value vectors (Section 3.1) associated with special tokens, periods, and commas tend to be small, canceling out the effect of large attention weights, thereby supporting the no-op hypothesis. Furthermore, it was shown that attention to the end-of-sequence token in MT models is used to ignore the contribution of the source sentence (Ferrando & Costa-jussà, 2021), useful when predicting some function words such as the particle “off” in “*She turned off the lights.*”. In auto-regressive LMs, these patterns are observed mainly in the beginning of sentence (BOS) token (Figure 13), although other tokens play the same role (Ferrando & Voita, 2024). According to Xiao et al. (2023), allowing attention mass on the BOS token is necessary for streaming generation, and performance degrades when the BOS is omitted. Using the logit spectroscopy (Section 4.3), Cancedda (2024) finds that early FFNs in Llama 2 write relevant information (for the attention sink mechanism to occur in later layers) into the residual stream of BOS. These FFNs write into the linear subspace spanned by the right singular vectors with the lowest associated singular values of the unembedding matrix. Cancedda (2024) refers to this as a *dark subspace* due to its low interference with next token prediction, and finds a significant correlation between the average attention received by a token and the existence of these dark signals in its residual stream. These dark signals reveal as massive activation values acting as fixed biases (Sun et al., 2024), a crucial prerequisite for the attention sink mechanism to take place (Puccetti et al., 2022; Bondarenko et al., 2023). On the other hand, specific neurons in the FFN of the layer before the attention head have been found to control the amount to which the tokens attend to BOS (Gurnee et al., 2024) (Figure 13).

Features in attention heads. Sparse autoencoders have been trained on the outputs of the attention layers to better understand the features computed by each head. Results presented in Kissane et al. (2024a) show that, on a two-layer Transformer, a large number of features (76%) are non-dead, with the majority of them being interpretable (82%). Three specific features are studied in detail. The “board by induction” feature promotes the token board, and is present on the output of an induction head (Section 5.1.2), being part of the **induction features** family. The “in questions starting with Which” feature is instead part of the **local context features**, promoting the prediction of ? when Which appears in the context. Lastly, “in texts related to pets” is an example of an **high-level context feature** that activates for almost the entire context, with its related head attending to pet-related context tokens. Notably, Kissane et al. (2024a) detect the presence of non-induction features in the output of induction-heads, providing evidence of **attention head polysemanticity**, initially observed by Heimersheim & Janiak (2023). Further investigations (Kissane et al., 2024b) reveal the same three feature families appear on GPT2-Small, as well as successor features, name mover features, suppression features and duplicate token features associated with heads matching their respective behaviors. Krzyzanowski et al. (2024) conduct a finer-grained analysis of features in GPT-2 Small attention heads, focusing on the top 10 features in each head and concluding that most heads do multiple tasks, with only around 10% of those being monosemantic. Their findings point out that early layers (0-3) mainly focus on shallow syntactic features, with the following layers encoding increasingly more complex syntactic features. Middle layers (5-6) contain the least interpretable features, while later layers (7-10) encode complex abstract features like time and distance relationships and high-level context concepts. The heads in the last attention block show mostly grammatical adjustments and bigram completions.

5.2 FEEDFORWARD NETWORK BLOCK

The dimensions in the FFN activation space (neurons), following the non-linearity, are more likely to be independently meaningful (Section 2.1.3), and have therefore been the object of study of recent interpretability works.

Neuron’s input behavior. The behavior of neurons in language models has been extensively studied, with examinations focusing on either their input or output behavior. In the context of input behavior analysis, Voita et al. (2023) show neurons firing exclusively on specific **position ranges**. Other discoveries include **skills neurons**, whose activations are correlated with the task of the input prompt (Wang et al., 2022), **concept-specific neurons** (Suau et al., 2020; 2022; Gurnee et al., 2023) whose response can be used to predict the presence of a concept in the provided context, such as whether it is Python code, French (Gurnee et al., 2023), or German (Quirke et al., 2023) language. Neurons responding to other linguistic and grammatical features have also been found (Bau et al., 2019; Durrani et al., 2023).

Neuron’s output behavior. Regarding the output behavior of neurons, Dai et al. (2022) use the Integrated Gradients method (Section 3.1) to attribute next-word facts predictions to FFNs neurons, finding **knowledge neurons**. The key-value memory perspective of FFNs (Section 2.1.3) offers a way to understand neuron’s weights. Specifically, using the direct logit attribution method (Section 3.2.1) we can measure the neuron’s effect on the logits. Geva et al. (2022b) show that some neurons promote the prediction of tokens associated with particular semantic and syntactic concepts. Ferrando et al. (2023) illustrate that a small set of neurons in later layers is responsible for making **linguistically acceptable predictions**, such as predicting the correct number of the verb, in agreement with the subject. Gurnee & Tegmark (2024) find neurons that interact with directions in the residual stream that are similar to the **space and time** feature directions extracted from probes. Tang et al. (2024) show **language-specific neurons** are key for multilingual generation, demonstrating one can steer the model output’s language by causally intervening on them. Finally, neurons **suppressing improbable continuations**, e.g. the repetition of the last token in the sequence, have recently been identified (Voita et al., 2023; Gurnee et al., 2024).

Polysemantic neurons. Recent work highlighted the presence of polysemantic neurons within language models. Notably, most early layer neurons specialize in sets of n-grams, functioning as **n-gram detectors** (Voita et al., 2023), with the majority of neurons firing on a large number of n-grams. Gurnee et al. (2023) suggest superposition appears in these early layers, and via sparse

probing they find sparse combinations of neurons whose added activation values disentangle the detection of specific n-grams, such as the compound word “social security” from other bigrams containing only one of the two terms. Even though polysemanticity and superposition arise in early layers, several **dead neurons** were observed in OPT models²⁰ (Voita et al., 2023). Furthermore, Elhage et al. (2022a) hypothesize models internally perform “**de-/re-tokenization**”, where neurons in early layers respond to multi-token words or compound words (Elhage et al., 2022a), mapping tokens to a more semantically meaningful representation (detokenization). In contrast, in the latest layers, neurons aggregate contextual representations back into single tokens (re-tokenization) to produce the next-token prediction.

Universality of neurons. Whether different models learn similar features remains an open question (Olah et al., 2020b). For instance, various computer vision models were found to learn Gabor filters in early layers (Olah et al., 2020a). In a recent study, Gurnee et al. (2024) investigated whether neurons respond to features similarly across different models. Their analysis used the pairwise correlation of neuron activations across GPT2 models trained from different random initializations as a proxy measure, revealing a subset of 1-5% of neurons activating on the same inputs. As expected, within the cluster of **universal neurons** there is a higher degree of monosemanticity. This group includes **alphabet neurons**, which activate in response to tokens representing individual letters and on tokens that start with the letter, supporting the re-tokenization hypothesis. Additionally, there are **previous token neurons** that fire based on the preceding token, as well as unigram, position, semantic, and syntax neurons. In terms of output behavior, universal neurons include **attention (de-)activation neurons**, responsible for controlling the amount of attention given to the BOS token by a subsequent attention head, and thus setting it as a no-op (Section 5.1.3). Lastly, Gurnee et al. (2024) hypothesize that some neurons act as **entropy neurons**, modulating the model’s uncertainty over the next token prediction.

High-level structure of the role of neurons. It has been suggested that the overall arrangement of neurons in language models mirrors that of neuroscience (Elhage et al., 2022a). Early layer neurons exhibit similarities to sensory neurons, responding to shallow patterns of the input, mostly focusing on n-grams. Moving into the middle layers, activation tends to occur around more high-level concepts (Bricken et al., 2023; Gurnee et al., 2023). An example of this is the neuron identified in Elhage et al. (2022a), which represents numbers only when they refer to the amount of people. Finally, later layers’ neurons bear a resemblance to motor neurons in the sense that they produce changes in the distribution of the next-token prediction, either by promoting or suppressing sets of tokens.

Features in Feedforward Networks. SAEs are able to identify significantly more interpretable features than the model’s neurons themselves (Bricken et al., 2023), as noted both by human and automated analyses in one-layer transformers. The features detected by SAEs trained to reconstruct FFN activations (Bricken et al., 2023) appear to split into increasingly more fine-grained distinctions of the feature as more dimensions (dictionary entries) are added, demonstrating that 512 neurons can encode tens of thousands of features. Examples of features found by Bricken et al. (2023) include those firing in the presence of Arabic or Hebrew scripts and promoting tokens in those scripts, and features responding to DNA sequences or base64 strings.

5.3 RESIDUAL STREAM

We can think of the residual stream as the main communication channel in a Transformer. The “direct path” (Section 2.2) connecting the input embedding with the unembedding matrix, $\mathbf{x}\mathbf{W}_U$ does not move information between positions, and mainly models bigram statistics (Elhage et al., 2021a), while the latest biases in the network, localized in the prediction head, are shown to shift predictions according to word frequency, promoting high-frequency tokens (Kobayashi et al., 2023). However, alternative paths involve the interaction between components, which write into linear subspaces (Elhage et al., 2021a) that can be read by downstream components, or directly by the prediction head, potentially doing more complex computations. Heimersheim & Turner (2023) observed that **the norm of the residual stream grows exponentially** along the layers over the forward pass of multiple Transformer LMs (Millidge & Winsor, 2023; Merrill et al., 2021). A similar growth rate appears

²⁰OPT models use ReLU activation functions, allowing for zero activation values (Zhang et al., 2022).

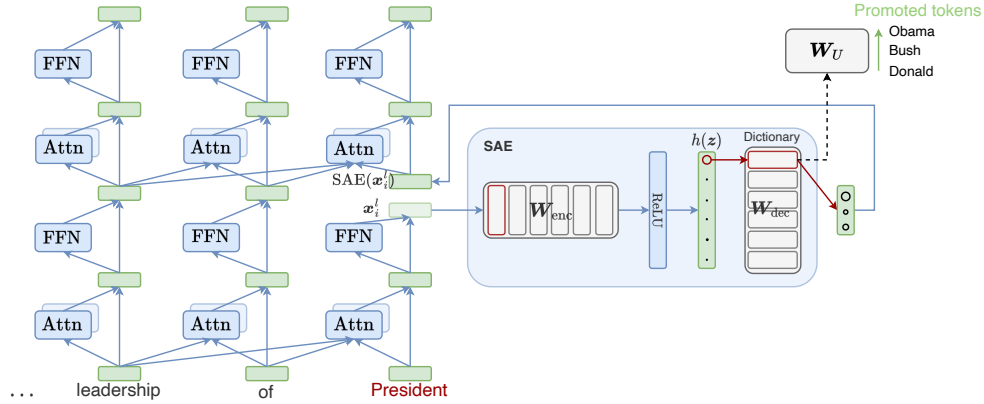


Figure 14: Example of the local context “President” feature in a Sparse Autoencoder (SAE) trained to reconstruct the second layer residual stream of GPT2-Small (Bloom, 2024).

in the norm of the output matrices writing into the residual stream, W_O and W_{out} , unlike input matrices (W_Q , W_K , W_V and W_{in}), which maintain constant norms along the layers. It is hypothesized that **some components perform memory management to remove information stored in the residual stream**. For instance, there are attention heads with OV matrices with negative eigenvalues attending to the current position, and FFN neurons whose input and output weights have large negative cosine similarity (Elhage et al., 2021a), meaning that they write a vector (FFN value) on the opposite direction to the direction they read from (FFN key). Notably, Gurnee et al. (2024) find that these neurons activate very frequently. Dao et al. (2023) evaluate a small Transformer LM and provide convincing evidence of multiple attention heads removing the information written by a first layer head.

Outlier dimensions (Kovaleva et al., 2021; Luo et al., 2021) have been identified within the residual stream. These **rogue dimensions exhibit large magnitudes relative to others and are associated with the generation of anisotropic representations** (Ethayarajh, 2019; Timkey & van Schijndel, 2021). Anisotropy means that the residual stream states of random pairs of tokens tend to point towards the same direction, i.e. the expected cosine similarity is close to one. Furthermore, ablating outlier dimensions has been shown to significantly decrease downstream performance (Kovaleva et al., 2021), suggesting they encode task-specific knowledge (Rudman et al., 2023). The magnitudes of these outliers have been shown to increase with model size (Dettmers et al., 2022), posing challenges for the quantization of large language models. The presence of rogue dimensions has been hypothesized to stem from optimizer choices (Elhage et al., 2023), with higher levels of regularization reducing their magnitudes (Ahmadian et al., 2023). Puccetti et al. (2022) identified a high correlation between the magnitude of the outlier dimensions found in token representations and their training frequency. They concluded that **these dimensions contribute to enabling the model to focus on special tokens, which is known to be associated with “no-op” attention updates** (Bondarenko et al., 2023) (see attention sinks in Section 5.1.3). In Vision Transformers, high-norm residual stream states have been identified as aggregators of global image information, appearing in patches with highly redundant information, such as those composing the image background (Darcet et al., 2024).

The specific features encoded within the residual stream at various layers remain uncertain, yet sparse autoencoders offer a promising avenue for improving our understanding. Recently, SAEs have been trained to reconstruct residual stream states in small language models such as GPT2-Small (Cunningham et al., 2023; Bloom & Lin, 2024; Bloom, 2024) showing highly interpretable features (Figure 14). Since residual stream states gather information about the sum of previous components’ outputs, inspecting SAE’s features can illuminate the process by which they are added or transformed during the forward pass. Given the type of features intermediate FFNs and attention heads interact with, we also expect the residual stream at middle layers to encode highly abstract features. Tigges et al. (2023) provide some preliminary evidence by showing that causally intervening on the residual stream in middle layers is more effective in flipping the sentiment of the output token, suggesting that the latent representation of sentiment is most prominent in the middle layers.

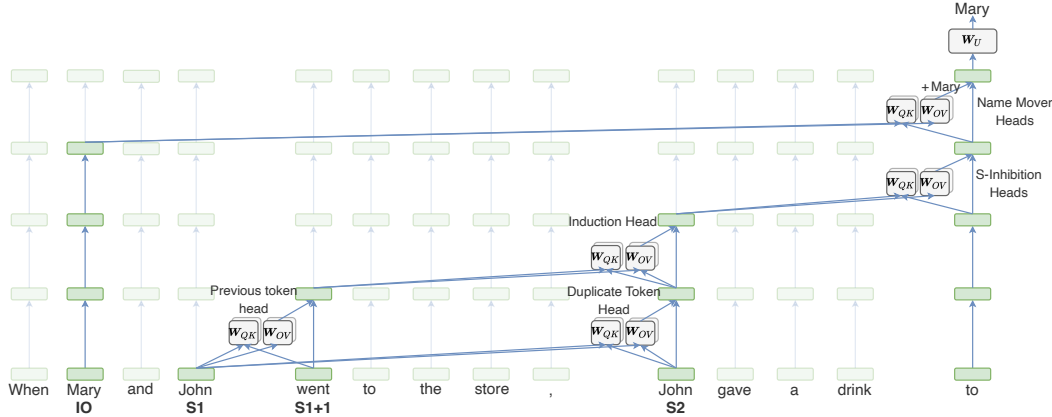


Figure 15: Simplified version of the IOI circuit in GPT2 Small discovered by Wang et al. (2023a).

Bloom & Lin (2024) study the features learned by a SAE in layer 8 of the 12-layer model GPT2-Small. Based on their output behavior via the logit lens (Section 4.3) the authors first find **local context features** promoting small sets of tokens. Secondly, they highlight the presence of **partition features**, which promote and suppress two distinct sets of tokens. For instance, a partition feature might promote tokens starting with capital letters and suppress those starting with lowercase letters. Finally, akin to suppression neurons (Voita et al., 2023; Gurnee et al., 2024), they note the presence of **suppression features** aimed at reducing the likelihood of specific sets of tokens. In line with these findings, recent studies have shown that language models create vectors representing functions or tasks given in-context examples (Hendel et al., 2023; Todd et al., 2024), which are found in intermediate layers. In the next section, we provide a deeper overview of the interaction between different components and the resulting behavior that emerges.

5.4 EMERGENT MULTI-COMPONENT BEHAVIORS

In previous sections we presented some of the different mechanisms that attention heads and FFNs implement, as well as an overview of the properties of the residual stream. However, in order to explain the remarkable performance of Transformers, we also need to account for the interactions between the different components (Wen et al., 2023; Cammarata et al., 2020).

Evidence of multi-component behavior. The induction mechanism presented in Section 5.1.2 is a clear example of two components (attention heads) composing together to complete a pattern. Recent evidence suggests that **multiple attention heads work together to create “function” or “task” vectors** describing the task when given in-context examples (Hendel et al., 2023; Todd et al., 2024). Intervening in the residual stream with those vectors can produce outputs in accordance with the encoded task on novel zero-shot prompts. Variengien & Winsor (2023) study in-context retrieval tasks involving answering a request where the answer can be found in the context. The authors identify a high-level mechanism that is universal across subtasks and models. Specifically, middle layers process the request, followed by a retrieval step of the entity from the context done by attention heads at later layers.

Additionally, Neo et al. (2024); Yu & Ananiadou (2024) reveal that **individual neurons within downstream FFNs activate according to the output of previous attention heads**, interacting in specific contexts. However, the most compelling evidence of particular behaviors emerging from the interaction between multiple components is found in the circuit analysis literature (Wang et al. (2023a); Stolfo et al. (2023b); Heimersheim & Janiak (2023); Geva et al. (2023); Hanna et al. (2023), among others). As an illustration, we present the circuit found in GPT2 Small for the Indirect Object Identification (IOI) task (Wang et al., 2023a), depicted in Figure 15. In the IOI task the model is given inputs of the type “*When Mary and John went to the store, John gave a drink to ____*”. The initial clause introduces two names (Mary and John), followed by a secondary clause where the two people exchange an item. The correct prediction is the name not appearing in the second clause, referred to as the Indirect Object (Mary). The circuit found in GPT2 Small mainly includes:

- Duplicity signaling: duplicate token heads at position S2, and an induction mechanism involving previous token heads at S1+1 signal the duplicity of S (John). This information is read by S-Inhibition heads at the last position, which write in the residual stream a token signal, indicating that S is repeated, and a position signal of the S1 token.
- Name copying: name mover heads in later layers copy information from names they attend to in the context to the last residual stream. However, the signals of the previous layers S-Inhibition heads modify the query of name mover heads so that the duplicated name (in S1 and S2) is less attended, favouring the copying of the Indirect Object (IO) and therefore, pushing its prediction.

Besides, Wang et al. (2023a) discovered Negative mover heads, which are instances of copy suppression heads (Section 5.1.2) downweighing the probability of the IO. While the IOI is an attention-centric circuit, examples of circuits involving both FFNs and attention heads are also present. For instance, Hanna et al. (2023) reverse-engineered the GPT2-Small circuit for the greater-than task, which involves sentences like *The war lasted from the year 1814 to the year 18__*, where the model must predict a year greater than 1814. The authors demonstrate that downstream FFNs compute a valid year by reading from previous attention heads, which attend to the event’s initial date.

Generality of circuits. Prakash et al. (2024) show that the **functionality of the circuit components remains consistent after fine-tuning** and benefits of fine-tuning are largely derived from an improved ability of circuit components to encode important task-relevant information rather than an overall functional rearrangement. Fine-tuned activations are also found to be compatible with the base model despite no explicit tuning constraints, suggesting the process produces minimal changes in the overall representation space. The findings of Prakash et al. (2024) are additionally supported by Jain et al. (2024) in controlled settings. While a common critique of mechanistic interpretability work is the limited scope of identified circuits, Merullo et al. (2024) show that low-level findings about specific heads and higher-level findings about general algorithms implemented by Transformer models can generalize across tasks, suggesting that large language models could be explained as functions of few task-general sparse components. The results of Merullo et al. (2024) also suggest that circuits are not *exclusive*, i.e. the same model components might be part of several circuits. Other studied dimensions of discovered circuits include their *faithfulness* (Hanna et al., 2024) and their *completeness* (Wang et al., 2023a).

Grokking as Emergence of Task-specific Circuits. Transformer models were observed to converge to different algorithmic solutions for tasks at hand (Zhong et al., 2023). Nanda et al. (2023a) provide convincing evidence on the relation between circuit emergence and *grokking*, i.e. the sudden emergence of near-perfect generalization capabilities for simple symbol manipulation tasks at late stages of model training (Power et al., 2022). Merrill et al. (2023) suggest the grokking phase transition can be seen as the emergence of a sparse circuit with generalization capabilities, replacing a dense subnetwork with low generalization capacity. According to Varma et al. (2023), this happens because dense memorizing circuits are inefficient for compressing large datasets. In contrast, generalizing circuits have a larger fixed cost but better per-example efficiency, hence being preferred in large-scale training. Huang et al. (2024b) connect the learning dynamic converging to grokking to the *double descent* phenomenon (Loog et al., 2020). According to this view, the emergence of specialized attention heads might be seen as a mild grokking-related phenomenon (Olsson et al., 2022; Bietti et al., 2023).

5.4.1 FACTUALITY AND HALLUCINATIONS IN MODEL PREDICTIONS

Intrinsic views on hallucinatory behavior. The generation of factually incorrect or nonsensical outputs is considered a significant limitation in the practical usage of language models (Ji et al., 2023; Minaee et al., 2024). While some techniques for detecting hallucinated content rely on quantifying the uncertainty of model predictions (Varshney et al., 2023), most alternative approaches engage with model internal representations. Approaches for detecting hallucinations directly from the representations include training probes and analyzing the properties of the representations leading to hallucinations. CH-Wang et al. (2023) and Azaria & Mitchell (2023) find probing classifiers predictive of the model’s output truthfulness, achieving the highest accuracy using middle and last layers representations. Zou et al. (2023) and Li et al. (2023a) find **“truthfulness” directions with**

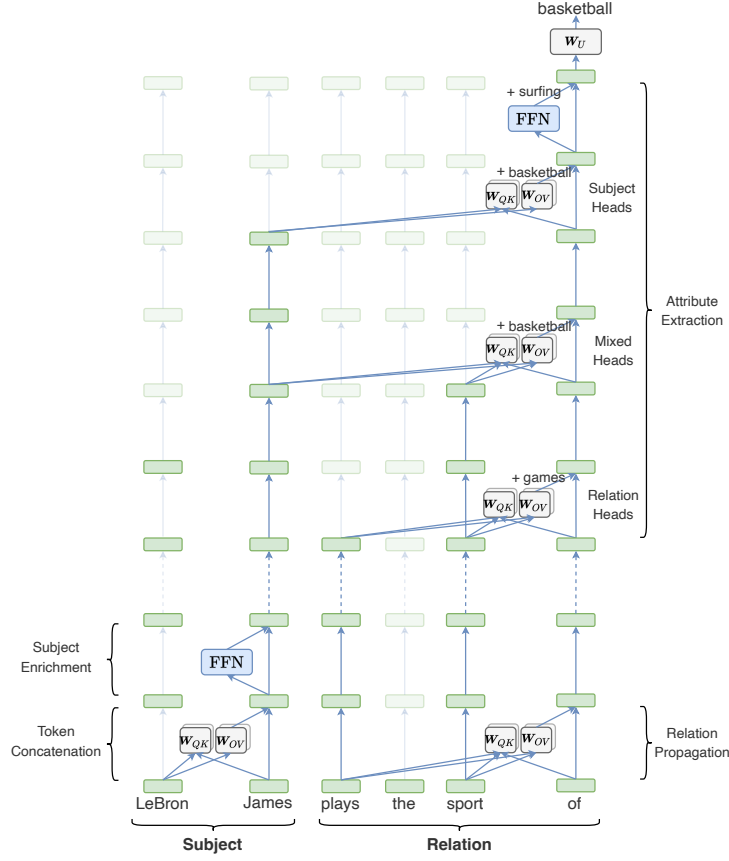


Figure 16: Simplified version of the factual recall circuit.

causal influence on the model outputs, i.e. intervening in the internal representations with the found directions enhance the output truthfulness. Li et al. (2023a) locate these causal directions in the specific attention head activation. Chen et al. (2024b) use the eigenvalues of responses’ representations covariance matrix to measure the semantic consistency in embedding space across layers, while Chen et al. (2024d) observe that logit lens (Section 4.3) scores of the predicted attribute (answer) in higher-layers representations of context tokens are informative of the answer correctness.

A related area of research with overlapping goals is that of hallucination detection in machine translation (MT). An MT model is considered to hallucinate if its output contains partially or fully detached content from the source sentence (Guerreiro et al., 2023b). Prediction probabilities of the generated sequence and attention distributions have been used to detect potential errors (Fomicheva et al., 2020) and model hallucinations (Guerreiro et al., 2023a;b). Recently, methods measuring the amount of contribution from the source sentence tokens (Ferrando et al., 2022a) were found to perform on par with external methods based on semantic similarity across several categories of model hallucinations (Dale et al., 2023a;b). Detection methods show complementary performance across hallucination categories, and simple aggregation strategies for internals-based detectors outperform methods relying on external semantic similarity or quality estimation modules (Himmi et al., 2024).

The underlying mechanisms involved in the prediction of hallucinated content for LLMs remain largely unexplained. Most of the research in this area focuses on studying the ability of language models to recall facts, which we discuss in the next section.

Recall of factual associations. Recent research has delved into the internal mechanisms through which language models recall factual information, which is directly related to the hallucination problem in LLMs. A common methodology involves studying tuples (s, r, a) , where s is a subject, r a relation, and a an attribute. The model is prompted to predict the attribute given the subject and relation. For instance, given the prompt: “*LeBron James plays the sport of*”, the model is

expected to predict *basketball*. Meng et al. (2022) and Geva et al. (2023) make use of causal interventions (Section 3.2.2) to localize a mechanism responsible for recalling factual knowledge within the language model. **Early-middle FFNs located in the last subject token add information about the subject** into its residual stream. On the other hand, information from the relation passes into the last token residual stream via early attention heads. Finally, **later layers attention heads extract the right attribute from the last subject residual stream**. Yuksekogonul et al. (2024) find that, in similar settings, attention to relevant tokens in the prompt correlates with LLM’s factual correctness. Importantly, the division of responsibilities between lower and upper layers was also observed in attention-less models based on the Mamba architecture (Gu & Dao, 2023; Sharma et al., 2024a). While this might be motivated by implicit context-mixing akin to Transformers’ causal self-attention (Ali et al., 2024), it suggests the organization of these mechanisms might be driven by the language modeling optimization process rather than architectural constraints.

Subsequent research has moved from localizing model behavior to studying the computations performed to solve this task. Hernandez et al. (2024) show that **attributes of entities can be linearly decoded from the enriched subject residual stream**, while Chughtai et al. (2024) investigate how attention heads’ OV circuits effectively decode the attributes, proposing an **additive mechanism**. More precisely, using the direct logit attribution by each token via the attention head (Equation (16)) they identify subject heads responsible for extracting attributes from the subject independently from the relation (not attending to it), as well as relation heads that promote attributes without being causally dependent on the subject. Additionally, a group of mixed heads generally favor the correct attribute and depend on both the subject and relation. The combination of the different heads’ outputs, each proposing different sets of attributes, together with the action of some downstream FFNs resolve the correct prediction (Figure 16). Nanda et al. (2023c) provide a detailed explanation of the subject enrichment phase by studying names of athletes as subjects. They suggest that the **first layers’ attention heads concatenate the athlete’s name on the final name token residual stream through addition**, and subsequent FFNs map the obtained athlete’s name representation into a linear representation of the athlete’s sport that can be easily linearly extracted by the downstream attribute extraction heads.

Merullo et al. (2023) report that for solving relational tasks, such as predicting a country’s capital given in-context examples, **middle layers prepare the argument**, e.g. Poland, of a `get_capital()` function that is applied downstream via an FFN update, giving place to `get_capital(Poland) = Warsaw`. Further research replicates Merullo et al. (2023)’s analysis on zero-shot settings (Lv et al., 2024) and finds specific attention heads “passing” the argument from the context (Poland), but also promoting the capital cities (Warsaw). Downstream FFNs “activate” relevant attention heads in the previous layer and add a vector guiding the residual stream toward the correct capital direction.

Recent works aim to shed light on how the model engages in factual recall vs. grounding. Following the aforementioned (subject, relation, attribute) structure of facts, an answer is considered to be grounded if the attribute is consistent with the information in the context of the prompt. Given prompts of the type “*The capital of Poland is London. Q: What is the capital of Poland? A:___*”, Yu et al. (2023a) find **in-context heads** and **memory heads** by using the difference logit attribution (Section 3.2.1, Equation (17)) of attention heads. These heads favor, respectively, the in-context answer *London* and the memorized answer *Warsaw*, showing a “competition” between mechanisms (Ortu et al., 2024). Furthermore, upweighting the output of each head type reveals a bias towards one of the two answers. Similar to the in-context heads, Variengien & Winsor (2023) show that a set of downstream attention heads retrieve the correct answer (an attribute) from the context via copying, preceded by a processing of the request (a question) in middle layers. Wu et al. (2024a) study these type of heads, which they coin **retrieval heads** in arbitrarily long-contexts, and show they are crucial for solving the Needle-in-a-Haystack tests (Kamradt, 2023). Monea et al. (2024) complement the findings of Yu et al. (2023a) and Meng et al. (2022) and show that **FFNs in the last token of the subject have higher contributions on ungrounded (memorized) answers as opposed to grounded answers**, while suggesting that grounding could be a more distributed process lacking a specific localization. Haviv et al. (2023) show that the recall of “memorized” idioms largely depends on the updates of the FFNs in early layers, providing further evidence of their role as a storage of memorized information. This is further observed in the study of memorized paragraphs, with lower layers exhibiting larger gradient flow (Stoeckl et al., 2024). On the other hand, Sharma et al. (2024b) show that substituting the original FFN matrices by lower-rank approximations (Sec-

tion 4.3) leads to improvements in model performance, especially in later layers of the model. They show that, in the factual recall task, the components with smaller singular values encode the correct semantic type of the answer but the wrong answer, thus their removal benefits the accuracy. To conclude, we draw a connection with a decoding strategy (DoLa) proposed to improve the factuality of language models (Chuang et al., 2024). DoLa contrastively compares the logit-lens next-token distributions between an early layer and a later layer (Li et al., 2023b), promoting tokens that undergo a larger probability change, suggesting that the factual knowledge injection is done in a distributed manner across the network.

Factuality issues and model editing. Factual information encoded in LMs might be incorrect from the start, or become obsolete over time. Moreover, inconsistencies have been observed when recalling factual knowledge in multilingual and cross-lingual settings (Fierro & Søgaard, 2022; Qi et al., 2023), or when factual associations are elicited using less common formulations (Berglund et al., 2023). This sparked the interest in developing *model editing* approaches able to perform targeted updates on model factual associations with minimal impact on other capabilities. While early approaches proposed edits based on external modules trained for knowledge editing (De Cao et al., 2021; Mitchell et al., 2022a;b), recent methods employ causal interventions (Section 3.2.2) to localize knowledge neurons (Dai et al., 2022) and FFNs in one or more layers (Meng et al., 2022; 2023), informed by factual recall mechanisms described in the previous paragraph. However, model editing approaches still present several challenges, summarized in (Yao et al., 2023; Li et al., 2024), including the risks of catastrophic forgetting (Gupta et al., 2024a;b) and downstream performance loss (Gu et al., 2024). Importantly, Hase et al. (2023) show that effective localization does not always result in improved editing results, and that distributed edits across different model sections can result in similar editing accuracy. Steerable-by-design architectures such as the Backpack Transformer (Hewitt et al., 2023) were recently proposed as possible alternatives to localization-driven methods, exploiting the linearity of component contributions (Section 4.2) as an inductive bias to enhance controllability. We refer readers to Wang et al. (2023b) for further insights on LM editing.

6 LM INTERPRETABILITY TOOLS

Several open-source software libraries were introduced to facilitate interpretability studies on Transformer-based LMs. In this section, we briefly summarize the most notable ones and highlight their main points of strength.

Input attribution tools. Captum (Kokhlikyan et al., 2020) is a library in the Pytorch ecosystem providing access to several gradient and perturbation-based input attribution methods for any Pytorch-based model. It notably supports training data attribution methods (Section 3.1), and recently added several utilities for simplifying attribution analyses of generative LMs (Miglani et al., 2023). Several Captum-based tools provide convenient APIs for input attribution of Transformer-based models: Transformers Interpret (Pierse, 2021), ferret (Attanasio et al., 2023) and Ecco (Alammar, 2021) are mainly centered around language classification tasks, while Inseq (Sarti et al., 2023) is focused specifically on generative LMs and supports advanced approaches for contrastive context attribution (Sarti et al., 2024) as well as context mixing evaluation (Section 3.1). SHAP (Lundberg & Lee, 2017) is a popular toolkit mainly centered on perturbation-based input attribution methods and model-agnostic explanations for various data modalities. The Saliency (PAIR Team, 2023) library provides framework-agnostic implementations for mainly gradient-based input attribution methods. LIT (Tenney et al., 2020) is a framework-agnostic tool providing a convenient set of utilities and an intuitive interface for interpretability studies spanning input attribution, concept-based explanations and counterfactual behavior evaluation. It notably includes a visual tool for debugging complex LLM prompts (Tenney et al., 2024).

Component importance analysis tools. Tools supporting work on circuit discovery and causal interventions play a fundamental role in mechanistic studies, balancing the complexity and model-specific nature of intervention-based methods with a broad support for various pre-trained LM architectures. TransformerLens (Nanda & Bloom, 2022) is a Pytorch-based toolkit to conduct mechanistic interpretability analyses of generative language models inspired by the closed-source Garçon library (Elhage et al., 2021b). The library reimplements popular Transformer LM architectures, preserving compatibility with the popular transformers library (Wolf et al., 2020) while also provid-

ing utilities such as hook points around model activations and attention head decomposition to facilitate custom interventions. NNSight (Fiotto-Kaufman, 2024) provides a Pytorch-compatible interface for interpretability analyses. Its usage is not restricted to Transformer models, but it provides utilities to streamline the usage of transformers checkpoints. Its main peculiarity is the ability to compile an *intervention graph* that can be processed through delayed execution, enabling the extraction of arbitrary internal information from large LMs hosted on remote servers. Pyvene (Wu et al., 2024c) is a Pytorch-based library supporting complex intervention schemes, such as trainable (Geiger et al., 2023a) and mid-training interventions (Geiger et al., 2022), alongside various model categories beyond Transformers. Notably, it supports the serialization of intervention schemes to simplify analyses and promotes reusability. Several tools are currently used for the development of SAEs (Section 4.2), providing overlapping sets of features. For example, SAELens (Bloom & Channin, 2024) supports advanced visualization of SAE features, while dictionary-learning (Marks & Mueller, 2023) is an actively developed tool built on top of NNSight, supporting various experimental features to address SAEs’ weaknesses. Finally, sparse-autoencoder (Cooney, 2023) provides a standard TransformerLens-compatible SAE implementation.

Tools for visualizing model internals. Several tools such as BERTViz (Vig, 2019), exBERT (Hoover et al., 2020) and Interpret (Lal et al., 2021) were developed to visualize attention weights and activations in Transformer-based LMs. LM-Debugger (Geva et al., 2022a) is a toolkit to inspect intermediate representation updates through the lens of logit attribution (Section 3.2.1), while VISIT (Katz & Belinkov, 2023), Ecco (Alammar, 2021) and Tuned Lens²¹ (Belrose et al., 2023a) simplify the application of naive and learned vocabulary projections to inspect the evolution of predictions across model layers. CircuitsVis (Cooney, 2022) provides reusable Python bindings for front-end components that can be used to visualize Transformers internals and predictions, and was adopted by various interpretability tools. Penzai (Daniel Johnson, 2024) is a JAX library supporting rich visualizations of pytree data structures, including LM weights and activations. LM-TT (Tufanov et al., 2024) allows inspecting the information flow in a forward pass, facilitating the examination of the contributions of individual attention heads and feed-forward neurons. TDB (Mossing et al., 2024) is a visual interface to interpret neuron activations in LMs supporting automated interpretability techniques and SAEs. Neuronpedia²² (Lin & Bloom, 2024) provides an open repository for visualizing activation of SAE features trained on LM residual stream states (Section 4.2). Notably, it includes a gamified experience to facilitate the annotation of human-interpretable concepts in SAE feature space. Lastly, sae-vis (McDougall & Bloom, 2024) is a SAELens-compatible library to produce feature-centric and prompt-centric interactive visualizations of SAE features.

Other notable interpretability-related tools. The “Restricted Access Sequence Processing Language” (RASP, Weiss et al., 2021) is a sequence processing language providing a human-readable model for transformer computations. Tracr (Lindner et al., 2023) is a compiler converting RASP programs into decoder-only Transformer weights, automating the creation of small Transformer models implementing specific desired behaviors. RASP and Tracr were adopted for promoting interpretable behaviors via constrained optimization (Friedman et al., 2023) and validating the effectiveness of circuit discovery techniques (Conmy et al., 2023). Pyreft²³ (Wu et al., 2024b) is a toolkit based on Pyvene for fine-tuning and sharing trainable interventions (Section 3.2.3, Causal abstraction) aimed at optimizing LM performance on selected tasks, in a similar but more targeted and efficient way than parameter-efficient fine-tuning methods (PEFT, Han et al., 2024). Going beyond the textual modality, ViT Prisma (Joseph, 2023) is a toolkit to conduct mechanistic interpretability analyses on vision and multimodal models. Finally, MAIA (Shaham et al., 2024) is a multimodal language model augmented with tool use to automate common interpretability workflows such as neuron explanations, example synthesis and counterfactual editing.

7 CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have offered an overview of the existing interpretability methods useful for understanding Transformer-based language models, and have presented the insights they have led to.

²¹<https://github.com/AlignmentResearch/tuned-lens>

²²<https://neuronpedia.org>

²³<https://github.com/stanfordnlp/pyreft>

Although the focus of this work is on practical methods and findings, we acknowledge theoretical studies related to the interpretability of Transformers, such as investigations explaining in-context learning (Akyürek et al., 2023; Von Oswald et al., 2023; Xie et al., 2022), explorations of Transformers through the lens of data compression and representation learning (Yu et al., 2023b; Voita et al., 2019a), the study of Transformers’ learning dynamics (Tian et al., 2024; 2023; Tarzanagh et al., 2024), or the analyses on their generalization properties on algorithmic tasks (Nogueira et al., 2021; Anil et al., 2022; Zhou et al., 2024).

Looking forward, we believe that the ultimate test for insights collected in years of interpretability work remains their applicability in debugging and improving the safety and reliability of future models, providing developers and users with better tools to interact with them and understand the factors influencing their predictions (Longo et al., 2024). To ensure such requirements are met, future developments in interpretability research will be faced with the challenging task of moving from *functionally-grounded evaluations* (i.e. no human evaluation, only toy settings) to actionable insights and benefits for real-world tasks (Doshi-Velez & Kim, 2017). From an analytical standpoint, this involves moving from methods and analyses operating in model component space to human-interpretable space, i.e from model components to features and natural language explanations, as suggested by Singh et al. (2024b), while still faithfully reflecting model behaviors (Siegel et al., 2024). Directions we deem promising in this area involve the usage of LMs as *verbalizers* (Feldhus et al., 2023; Bills et al., 2023; Wang et al., 2024; Chen et al., 2024c) for scaling input and component attribution analyses, especially when paired with verification mechanisms to ensure counterfactual consistency (Avitan et al., 2024), and circuit discovery methods leveraging interpretable features to enable interventions motivated by human-understandable concepts (Marks et al., 2024). More accessible insights might also unlock gains in model performance and efficiency, translating interpretability-driven insights into downstream task improvements (Wu et al., 2024b). Importantly, interdisciplinary research grounded in the technical developments we summarize in this survey will play a key role in broadening the scope of interpretability analyses to account for the perceptual and interactive dimensions of model explanations from a human perspective (Liao et al., 2020; Dhanorkar et al., 2021; Vasconcelos et al., 2023). Ultimately, we believe that ensuring open and convenient access to the internals of advanced LMs will remain a fundamental prerequisite for future progress in this area (Bau et al., 2023; Casper et al., 2024; Hudson et al., 2024).

ACKNOWLEDGEMENTS

Javier Ferrando is supported by the Spanish Ministerio de Ciencia e Innovación through the project PID2019-107579RB-I00 / AEI / 10.13039/501100011033. Gabriele Sarti and Arianna Bisazza acknowledge the support of the Dutch Research Council (NWO) as part of the project InDeep (NWA.1292.19.399).

REFERENCES

- S. Abnar and W. Zuidema. Quantifying attention flow in transformers. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.385. URL <https://aclanthology.org/2020.acl-main.385>. (p. 7)
- R. Achitbat, S. M. V. Hatefi, M. Dreyer, A. Jain, T. Wiegand, S. Lapuschkin, and W. Samek. AttnLRP: Attention-aware layer-wise relevance propagation for transformers, 2024. (p. 7)
- J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, pp. 9525–9536, Red Hook, NY, USA, 2018. Curran Associates Inc. (p. 8)
- J. Adebayo, M. Muelly, I. Liccardi, and B. Kim. Debugging tests for model explanations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20*, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546. (p. 8)
- J. Adebayo, M. Muelly, H. Abelson, and B. Kim. Post hoc explanations may be ineffective for detecting unknown spurious correlation. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=xNOVFCCvDpM>. (p. 8)
- C. Agarwal, S. H. Tanneru, and H. Lakkaraju. Faithfulness vs. plausibility: On the (un)reliability of explanations from large language models. *ArXiv*, abs/2402.04614, 2024. URL <https://api.semanticscholar.org/CorpusID:267523276>. (p. 19)

-
- A. Ahmadian, S. Dash, H. Chen, B. Venkitesh, Z. S. Gou, P. Blunsom, A. Üstün, and S. Hooker. Intriguing properties of quantization at scale. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=IYe8j7Gy8f>. (p. 24)
- E. Akyurek, T. Bolukbasi, F. Liu, B. Xiong, I. Tenney, J. Andreas, and K. Guu. Towards tracing knowledge in language models back to the training data. In Y. Goldberg, Z. Kozareva, and Y. Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2429–2446, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.180. URL <https://aclanthology.org/2022.findings-emnlp.180>. (p. 8)
- E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=g0X4H8yN4I>. (p. 31)
- E. Akyürek, B. Wang, Y. Kim, and J. Andreas. In-context language learning: Architectures and algorithms, 2024. (p. 20)
- G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. *Arxiv*, 2016. URL <https://arxiv.org/abs/1610.01644>. (p. 13)
- J. Alammr. Ecco: An open source library for the explainability of transformer language models. In H. Ji, J. C. Park, and R. Xia (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pp. 249–257, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-demo.30. URL <https://aclanthology.org/2021.acl-demo.30>. (p. 29, 30)
- A. Ali, T. Schnake, O. Eberle, G. Montavon, K.-R. Müller, and L. Wolf. XAI for transformers: Better explanations through conservative propagation. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 435–451. PMLR, 17–23 Jul 2022. URL <https://proceedings.mlr.press/v162/ali22a.html>. (p. 7)
- A. Ali, I. Zimmerman, and L. Wolf. The hidden attention of mamba models, 2024. (p. 28)
- K. Amara, R. Sevastjanova, and M. El-Assady. Syntaxshap: Syntax-aware explainability method for text generation. *Arxiv*, abs/2402.09259, 2024. URL <https://api.semanticscholar.org/CorpusID:267657673>. (p. 7)
- C. Anil, Y. Wu, A. J. Andreassen, A. Lewkowycz, V. Misra, V. V. Ramasesh, A. Slone, G. Gur-Ari, E. Dyer, and B. Neyshabur. Exploring length generalization in large language models. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=zSkYVeX7bC4>. (p. 31)
- Anonymous. The disagreement problem in explainable machine learning: A practitioner’s perspective. *Submitted to Transactions on Machine Learning Research*, 2024. URL <https://openreview.net/forum?id=jESY2WTZCe>. Under review. (p. 8)
- A. Ardit, O. Balcells, A. Syed, W. Gurnee, and N. Nanda. Refusal in llms is mediated by a single direction. *Alignment Forum*, 2024. URL <https://alignmentforum.org/posts/jGuXSZgv6qfdhMCuJ/refusal-in-llms-is-mediated-by-a-single-direction>. (p. 15)
- A. Arora, D. Jurafsky, and C. Potts. Causalgym: Benchmarking causal interpretability methods on linguistic tasks, 2024. URL <https://arxiv.org/abs/2402.12560>. (p. 12, 15)
- S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018. doi: 10.1162/tacl_a_00034. URL <https://aclanthology.org/Q18-1034>. (p. 15)
- P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. A diagnostic study of explainability techniques for text classification. In B. Webber, T. Cohn, Y. He, and Y. Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3256–3274, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.263. URL <https://aclanthology.org/2020.emnlp-main.263>. (p. 8)
- P. Atanasova, O.-M. Camburu, C. Lioma, T. Lukasiewicz, J. G. Simonsen, and I. Augenstein. Faithfulness tests for natural language explanations. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 283–294, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.25. URL <https://aclanthology.org/2023.acl-short.25>. (p. 18)
- G. Attanasio, E. Pastor, C. Di Bonaventura, and D. Nozza. ferret: a framework for benchmarking explainers on transformers. In D. Croce and L. Soldaini (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 256–266, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-demo.29. URL <https://aclanthology.org/2023.eacl-demo.29>. (p. 29)
- M. Avitan, R. Cotterell, Y. Goldberg, and S. Ravfogel. What changed? converting representational interventions to natural language. *Arxiv*, 2024. URL <https://arxiv.org/abs/2402.11355>. (p. 31)
- A. Azaria and T. Mitchell. The internal state of an LLM knows when it’s lying. In H. Bouamor, J. Pino, and K. Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.68. URL <https://aclanthology.org/2023.findings-emnlp.68>. (p. 26)

- J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *Arxiv*, 2016. URL <https://arxiv.org/abs/1607.06450>. (p. 3)
- S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015. doi: 10.1371/journal.pone.0130140. URL <https://doi.org/10.1371/journal.pone.0130140>. (p. 7)
- Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, and J. Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv*, 2022. (p. 15)
- D. Balduzzi, M. Frean, L. Leary, J. P. Lewis, K. W.-D. Ma, and B. McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In D. Precup and Y. W. Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 342–350. PMLR, 06–11 Aug 2017. URL <https://proceedings.mlr.press/v70/balduzzi17b.html>. (p. 6)
- J. Bastings and K. Filippova. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In A. Alishahi, Y. Belinkov, G. Chrupala, D. Hupkes, Y. Pinter, and H. Sajjad (eds.), *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 149–155, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.14. URL <https://aclanthology.org/2020.blackboxnlp-1.14>. (p. 7)
- J. Bastings, S. Ebert, P. Zablotskaia, A. Sandholm, and K. Filippova. “will you find these shortcuts?” a protocol for evaluating the faithfulness of input salience methods for text classification. In Y. Goldberg, Z. Kozareva, and Y. Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 976–991, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.64. URL <https://aclanthology.org/2022.emnlp-main.64>. (p. 6)
- A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, and J. Glass. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=H1z-PsR5KX>. (p. 22)
- D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020. doi: 10.1073/pnas.1907375117. URL <https://www.pnas.org/doi/abs/10.1073/pnas.1907375117>. (p. 14)
- D. Bau, B. C. Wallace, A. Guha, J. Bell, and C. Brodley. National deep inference facility for very large language models (ndif). *United States National Science Foundation*, 2023. (p. 31)
- Y. Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1): 207–219, March 2022. doi: 10.1162/coli_a_00422. URL <https://aclanthology.org/2022.cl-1.7>. (p. 13, 14)
- Y. Belinkov and J. Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, 2019. doi: 10.1162/tac1_a_00254. URL <https://aclanthology.org/Q19-1004>. (p. 14)
- Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, and J. Glass. What do neural machine translation models learn about morphology? In R. Barzilay and M.-Y. Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 861–872, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1080. URL <https://aclanthology.org/P17-1080>. (p. 13)
- N. Belrose. Least-squares concept erasure with oracle concept labels. *EleutherAI Blog*, 2023. URL <https://blog.eleuther.ai/oracle-leace/>. (p. 14)
- N. Belrose, Z. Furman, L. Smith, D. Halawi, I. Ostrovsky, L. McKinney, S. Biderman, and J. Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *Arxiv*, 2023a. URL <https://arxiv.org/abs/2303.08112>. (p. 17, 30)
- N. Belrose, D. Schneider-Joseph, S. Ravfogel, R. Cotterell, E. Raff, and S. Biderman. LEACE: Perfect linear concept erasure in closed form. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL <https://openreview.net/forum?id=awIpKpwTWF>. (p. 14)
- Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, mar 2003. ISSN 1532-4435. (p. 2)
- L. Bereska and E. Gavves. Mechanistic interpretability for ai safety – a review. *ArXiv*, 2024. URL <https://arxiv.org/abs/2404.14082>. (p. 2)
- L. Berglund, M. Tong, M. Kaufmann, M. Balesni, A. C. Stickland, T. Korbak, and O. Evans. The reversal curse: Llms trained on “a is b” fail to learn “b is a”. *ArXiv*, abs/2309.12288, 2023. URL <https://api.semanticscholar.org/CorpusID:262083829>. (p. 29)
- A. Bibal, R. Cardon, D. Alfter, R. Wilkens, X. Wang, T. François, and P. Watrin. Is attention explanation? an introduction to the debate. In S. Muresan, P. Nakov, and A. Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3889–3900, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.269. URL <https://aclanthology.org/2022.acl-long.269>. (p. 19)

- S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, and O. Van Der Wal. Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023. (p. 21)
- A. Bietti, V. Cabannes, D. Bouchacourt, H. Jegou, and L. Bottou. Birth of a transformer: A memory viewpoint. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 1560–1588. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/0561738a239a995c8cd2ef0e50cfa4fd-Paper-Conference.pdf. (p. 26)
- S. Bills, N. Cammarata, D. Mossing, H. Tillman, L. Gao, G. Goh, I. Sutskever, J. Leike, J. Wu, and W. Saunders. Language models can explain neurons in language models. <https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html>, 2023. (p. 18, 31)
- B. Bilodeau, N. Jaques, P. W. Koh, and B. Kim. Impossibility theorems for feature attribution. *Proceedings of the National Academy of Sciences*, 121(2):e2304406120, 2024. doi: 10.1073/pnas.2304406120. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2304406120>. (p. 8)
- J. Bloom. Open source sparse autoencoders for all residual stream layers of GPT2 small. *AI Alignment Forum*, 2024. URL <https://www.alignmentforum.org/posts/f9EgflSurAiqRJySD/open-source-sparse-autoencoders-for-all-residual-stream>. (p. 24)
- J. Bloom and D. Channin. Saelens. *GitHub repository*, 2024. URL <https://github.com/jbloomAus/SAELens>. (p. 30)
- J. Bloom and J. Lin. Understanding SAE features with the logit lens. *AI Alignment Forum*, 2024. URL <https://www.alignmentforum.org/posts/qykrYY6rXXM7EEs8Q/understanding-sae-features-with-the-logit-lens>. (p. 24, 25)
- T. Bolukbasi, A. Pearce, A. Yuan, A. Coenen, E. Reif, F. Viégas, and M. Wattenberg. An interpretability illusion for bert, 2021. (p. 18)
- Y. Bondarenko, M. Nagel, and T. Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=sbusw6LD41>. (p. 21, 24)
- T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, Z. Hatfield-Dodds, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL <https://transformer-circuits.pub/2023/monosemantic-features/index.html>. (p. 15, 16, 23, 54, 55)
- S. Brody, U. Alon, and E. Yahav. On the expressivity role of LayerNorm in transformers' attention. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 14211–14221, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.895. URL <https://aclanthology.org/2023.findings-acl.895>. (p. 3)
- D. Brown, N. Vyas, and Y. Bansal. On privileged and convergent bases in neural network representations, 2023. (p. 5)
- T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfbcb4967418bfb8ac142f64a-Paper.pdf. (p. 2)
- M.-E. Brunet, C. Alkalay-Houlihan, A. Anderson, and R. Zemel. Understanding the origins of bias in word embeddings. In K. Chaudhuri and R. Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 803–811. PMLR, 09–15 Jun 2019. URL <https://proceedings.mlr.press/v97/brunet19a.html>. (p. 8)
- G. Brunner, Y. Liu, D. Pascual, O. Richter, M. Ciaramita, and R. Wattenhofer. On identifiability in transformers. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=BJg1f6EFDB>. (p. 7)
- C. Burns, H. Ye, D. Klein, and J. Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=ETKGuby0hcs>. (p. 14)
- N. Cammarata, S. Carter, G. Goh, C. Olah, M. Petrov, L. Schubert, C. Voss, B. Egan, and S. K. Lim. Thread: Circuits. *Distill*, 2020. doi: 10.23915/distill.00024. URL <https://distill.pub/2020/circuits>. (p. 11, 25)
- N. Cancedda. Spectral filters, dark signals, and attention sinks, 2024. URL <https://arxiv.org/abs/2402.09221>. (p. 18, 21)
- S. Casper, C. Ezell, C. Siegmann, N. Kolt, T. L. Curtis, B. Bucknall, A. A. Haupt, K. Wei, J. Scheurer, M. Hobbhahn, L. Sharkey, S. Krishna, M. von Hagen, S. Alberti, A. Chan, Q. Sun, M. Gerovitch, D. Bau, M. Tegmark, D. Krueger, and D. Hadfield-Menell. Black-box access is insufficient for rigorous ai audits. *ArXiv*, abs/2401.14446, 2024. URL <https://api.semanticscholar.org/CorpusID:267301601>. (p. 31)

- S. CH-Wang, B. V. Durme, J. Eisner, and C. Kedzie. Do androids know they're only dreaming of electric sheep?, 2023. URL <https://arxiv.org/abs/2312.17249v1>. (p. 14, 26)
- L. Chan, A. Garriga-Alonso, N. Goldwosky-Dill, R. Greenblatt, J. Nitishinskaya, A. Radhakrishnan, B. Shlegeris, and N. Thomas. Causal scrubbing, a method for rigorously testing interpretability hypotheses. *AI Alignment Forum*, 2022. URL <https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN/causal-scrubbing-a-method-for-rigorously-testing>. (p. 10)
- H. Chefer, S. Gur, and L. Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 782–791, June 2021. (p. 7)
- A. Chen, R. Shwartz-Ziv, K. Cho, M. L. Leavitt, and N. Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*, 2024a. URL <https://openreview.net/forum?id=M05PiKHELW>. (p. 19)
- C. Chen, K. Liu, Z. Chen, Y. Gu, Y. Wu, M. Tao, Z. Fu, and J. Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024b. URL <https://openreview.net/forum?id=Zj12nzlQbz>. (p. 27)
- H. Chen, C. Vondrick, and C. Mao. Selfie: Self-interpretation of large language model embeddings, 2024c. URL <https://arxiv.org/abs/2403.10949>. (p. 17, 31)
- S. Chen, M. Xiong, J. Liu, Z. Wu, T. Xiao, S. Gao, and J. He. In-context sharpness as alerts: An inner representation perspective for hallucination mitigation, 2024d. (p. 27)
- A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. URL <http://jmlr.org/papers/v24/22-1144.html>. (p. 2, 4)
- A. G. Chowdhury, M. M. Islam, V. Kumar, F. H. Shezan, V. Kumar, V. Jain, and A. Chadha. Breaking down the defenses: A comparative survey of attacks on large language models, 2024. (p. 18)
- Y.-S. Chuang, Y. Xie, H. Luo, Y. Kim, J. R. Glass, and P. He. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=Th6NyL07na>. (p. 29)
- B. Chughtai, A. Cooney, and N. Nanda. Summing up the facts: Additive mechanisms behind factual recall in llms, 2024. URL <https://www.arxiv.org/abs/2402.07321>. (p. 21, 28)
- K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? an analysis of BERT's attention. In T. Linzen, G. Chrupala, Y. Belinkov, and D. Hupkes (eds.), *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL <https://aclanthology.org/W19-4828>. (p. 19, 21)
- T. Conerly, A. Templeton, T. Bricken, J. Marcus, and T. Henighan. Circuits updates - april 2024. update on how we train saes. *Transformer Circuits Thread*, 2024. URL <https://transformer-circuits.pub/2024/april-update/index.html>. (p. 55)
- A. Conmy, A. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 16318–16352. Curran Associates, Inc., 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Abstract-Conference.html. (p. 10, 12, 30)
- A. Cooney. CircuitVis, December 2022. URL <https://github.com/alan-cooney/CircuitVis>. (p. 30)
- A. Cooney. Sparse autoencoder. *GitHub repository*, 2023. URL https://github.com/ai-safety-foundation/sparse_autoencoder. (p. 30)
- G. M. Correia, V. Niculae, and A. F. T. Martins. Adaptively sparse transformers. In K. Inui, J. Jiang, V. Ng, and X. Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2174–2184, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1223. URL <https://aclanthology.org/D19-1223>. (p. 19)
- M. Costa-jussà, E. Smith, C. Ropers, D. Licht, J. Maillard, J. Ferrando, and C. Escolano. Toxicity in multilingual machine translation at scale. In H. Bouamor, J. Pino, and K. Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9570–9586, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.642. URL <https://aclanthology.org/2023.findings-emnlp.642>. (p. 2)
- I. Covert, S. Lundberg, and S.-I. Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021. URL <http://jmlr.org/papers/v22/20-1316.html>. (p. 7)

-
- J. Crabbé and M. van der Schaar. Evaluating the robustness of interpretability methods through explanation invariance and equivariance. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=5UwnKSgY6u>. (p. 8)
- R. Csordás, S. van Steenkiste, and J. Schmidhuber. Are neural nets modular? inspecting functional modularity through differentiable weight masks. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=7uVcpu-gMD>. (p. 10)
- H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey. Sparse autoencoders find highly interpretable features in language models. *Arxiv*, 2023. URL <https://arxiv.org/abs/2309.08600>. (p. 15, 24)
- D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, and F. Wei. Knowledge neurons in pretrained transformers. In S. Muresan, P. Nakov, and A. Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.581. URL <https://aclanthology.org/2022.acl-long.581>. (p. 22, 29)
- D. Dale, E. Voita, L. Barrault, and M. R. Costa-jussà. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity Even better. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36–50, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.3. URL <https://aclanthology.org/2023.acl-long.3>. (p. 27)
- D. Dale, E. Voita, J. Lam, P. Hansanti, C. Ropers, E. Kalbassi, C. Gao, L. Barrault, and M. Costa-jussà. HalOm: A manually annotated benchmark for multilingual hallucination and omission detection in machine translation. In H. Bouamor, J. Pino, and K. Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 638–653, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.42. URL <https://aclanthology.org/2023.emnlp-main.42>. (p. 27)
- F. Dalvi, N. Durrani, H. Sajjad, Y. Belinkov, A. Bau, and J. Glass. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI’19/IAAI’19/EAAI’19*. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33016309. URL <https://doi.org/10.1609/aaai.v33i01.33016309>. (p. 18)
- P. A. Daniel Johnson. Penzai. *GitHub repository*, 2024. URL <https://github.com/google-deeppmind/penzai>. (p. 30)
- J. Dao, Y.-T. Lau, C. Rager, and J. Janiak. An adversarial example for direct logit attribution: Memory management in gelu-4l, 2023. (p. 24)
- G. Dar, M. Geva, A. Gupta, and J. Berant. Analyzing transformers in embedding space. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16124–16170, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.893. URL <https://aclanthology.org/2023.acl-long.893>. (p. 17, 18)
- T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=2dn03LLiJ1>. (p. 24)
- Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML’17*, pp. 933–941. JMLR.org, 2017. (p. 16)
- N. De Cao, M. S. Schlichtkrull, W. Aziz, and I. Titov. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In B. Webber, T. Cohn, Y. He, and Y. Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3243–3255, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.262. URL <https://aclanthology.org/2020.emnlp-main.262>. (p. 10)
- N. De Cao, W. Aziz, and I. Titov. Editing factual knowledge in language models. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.522. URL <https://aclanthology.org/2021.emnlp-main.522>. (p. 29)
- N. De Cao, L. Schmid, D. Hupkes, and I. Titov. Sparse interventions in language models with differentiable masking. In J. Bastings, Y. Belinkov, Y. Elazar, D. Hupkes, N. Saphra, and S. Wiegrefe (eds.), *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 16–27, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.blackboxnlp-1.2. URL <https://aclanthology.org/2022.blackboxnlp-1.2>. (p. 10)
- B. Deiseroth, M. Deb, S. Weinbach, M. Brack, P. Schramowski, and K. Kersting. Atman: Understanding transformer predictions through memory efficient attention manipulation. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 63437–63460. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c83bc020a020cdeb966ed10804619664-Paper-Conference.pdf. (p. 7)

- M. Denil, A. Demiraj, and N. de Freitas. Extraction of salient sentences from labelled documents. *Arxiv*, 2015. URL <https://arxiv.org/abs/1412.6815>. (p. 6)
- T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 30318–30332. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c3ba4962c05c49636d4c6206a97e9c8a-Paper-Conference.pdf. (p. 24)
- J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio (eds.), *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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>. (p. 2, 14)
- J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4443–4458, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL <https://aclanthology.org/2020.acl-main.408>. (p. 7)
- K. Dhamdhere, M. Sundararajan, and Q. Yan. How important is a neuron. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=SylKoo0cKm>. (p. 12)
- S. Dhanorkar, C. T. Wolf, K. Qian, A. Xu, L. Popa, and Y. Li. Who needs to know what, when?: Broadening the explainable ai (xai) design space by looking at explanations across the ai lifecycle. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference, DIS '21*, pp. 1591–1602, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384766. doi: 10.1145/3461778.3462131. URL <https://doi.org/10.1145/3461778.3462131>. (p. 31)
- A. Y. Din, T. Karidi, L. Choshen, and M. Geva. Jump to conclusions: Short-cutting transformers with linear transformations. *Arxiv*, 2023. URL <https://arxiv.org/abs/2303.09435>. (p. 17)
- S. Ding and P. Koehn. Evaluating saliency methods for neural language models. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5034–5052, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.399. URL <https://aclanthology.org/2021.naacl-main.399>. (p. 6)
- F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning, 2017. URL <https://arxiv.org/abs/1702.08608>. (p. 31)
- N. Durrani, F. Dalvi, and H. Sajjad. Discovering salient neurons in deep nlp models. *Journal of Machine Learning Research*, 24(362):1–40, 2023. URL <http://jmlr.org/papers/v24/23-0074.html>. (p. 22)
- Y. Elazar, S. Ravfogel, A. Jacovi, and Y. Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 03 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00359. URL https://doi.org/10.1162/tacl_a_00359. (p. 14)
- N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021a. URL <https://transformer-circuits.pub/2021/framework/index.html>. (p. 2, 3, 6, 19, 20, 23, 24)
- N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. Garcon. *Transformer Circuits Thread*, 2021b. URL <https://transformer-circuits.pub/2021/garcon/index.html>. (p. 29)
- N. Elhage, T. Hume, C. Olsson, N. Nanda, T. Henighan, S. Johnston, S. ElShowk, N. Joseph, N. DasSarma, B. Mann, D. Hernandez, A. Askell, K. Ndousse, A. Jones, D. Drain, A. Chen, Y. Bai, D. Ganguli, L. Lovitt, Z. Hatfield-Dodds, J. Kernion, T. Conerly, S. Kravec, S. Fort, S. Kadavath, J. Jacobson, E. Tran-Johnson, J. Kaplan, J. Clark, T. Brown, S. McCandlish, D. Amodei, and C. Olah. Softmax linear units. *Transformer Circuits Thread*, 2022a. URL <https://transformer-circuits.pub/2022/solu/index.html>. (p. 23)
- N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, and C. Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022b. URL https://transformer-circuits.pub/2022/toy_model/index.html. (p. 5, 15)
- N. Elhage, R. Lasenby, and C. Olah. Privileged bases in the transformer residual stream. *Transformer Circuits Thread*, 2023. URL <https://transformer-circuits.pub/2023/privileged-basis/index.html>. (p. 24)
- J. Enguehard. Sequential integrated gradients: a simple but effective method for explaining language models. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 7555–7565, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.477. URL <https://aclanthology.org/2023.findings-acl.477>. (p. 7)

-
- K. Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In K. Inui, J. Jiang, V. Ng, and X. Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1006. URL <https://aclanthology.org/D19-1006>. (p. 24)
- K. Ethayarajh and D. Jurafsky. Attention flows are shapley value explanations. In C. Zong, F. Xia, W. Li, and R. Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 49–54, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.8. URL <https://aclanthology.org/2021.acl-short.8>. (p. 7)
- N. Feldhus, L. Hennig, M. D. Nasert, C. Ebert, R. Schwarzenberg, and S. Möller. Saliency map verbalization: Comparing feature importance representations from model-free and instruction-based methods. In B. Dalvi Mishra, G. Durrett, P. Jansen, D. Neves Ribeiro, and J. Wei (eds.), *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pp. 30–46, Toronto, Canada, June 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.nlrse-1.4. URL <https://aclanthology.org/2023.nlrse-1.4>. (p. 31)
- J. Ferrando and M. R. Costa-jussà. Attention weights in transformer NMT fail aligning words between sequences but largely explain model predictions. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 434–443, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.39. URL <https://aclanthology.org/2021.findings-emnlp.39>. (p. 21)
- J. Ferrando and E. Voita. Information flow routes: Automatically interpreting language models at scale. *Arxiv*, 2024. URL <https://arxiv.org/abs/2403.00824>. (p. 4, 12, 19, 21)
- J. Ferrando, G. I. Gállego, B. Alastruey, C. Escolano, and M. R. Costa-jussà. Towards opening the black box of neural machine translation: Source and target interpretations of the transformer. In Y. Goldberg, Z. Kozareva, and Y. Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8756–8769, Abu Dhabi, United Arab Emirates, December 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.599. URL <https://aclanthology.org/2022.emnlp-main.599>. (p. 27)
- J. Ferrando, G. I. Gállego, and M. R. Costa-jussà. Measuring the mixing of contextual information in the transformer. In Y. Goldberg, Z. Kozareva, and Y. Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8698–8714, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.595. URL <https://aclanthology.org/2022.emnlp-main.595>. (p. 7)
- J. Ferrando, G. I. Gállego, I. Tsiamas, and M. R. Costa-jussà. Explaining how transformers use context to build predictions. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5486–5513, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.301. URL <https://aclanthology.org/2023.acl-long.301>. (p. 9, 22)
- C. Fierro and A. Søgaard. Factual consistency of multilingual pretrained language models. In S. Muresan, P. Nakov, and A. Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3046–3052, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.240. URL <https://aclanthology.org/2022.findings-acl.240>. (p. 29)
- J. Fiotto-Kaufman. nnsight: The package for interpreting and manipulating the internals of deep learned models., 2024. URL <https://github.com/JadenFiotto-Kaufman/nnsight>. (p. 30)
- M. Fomicheva, S. Sun, L. Yankovskaya, F. Blain, F. Guzmán, M. Fishel, N. Aletras, V. Chaudhary, and L. Specia. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555, 2020. doi: 10.1162/tacl_a_00330. URL <https://aclanthology.org/2020.tacl-1.35>. (p. 27)
- D. Friedman, A. Wettig, and D. Chen. Learning transformer programs. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. URL <https://openreview.net/forum?id=Pe9WxkN8Ff>. (p. 30)
- A. Geiger, K. Richardson, and C. Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In A. Alishahi, Y. Belinkov, G. Chrupala, D. Hupkes, Y. Pinter, and H. Sajjad (eds.), *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 163–173, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL <https://aclanthology.org/2020.blackboxnlp-1.16>. (p. 9)
- A. Geiger, H. Lu, T. Icard, and C. Potts. Causal abstractions of neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 9574–9586. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/4f5c422f4d49a5a807eda27434231040-Paper.pdf. (p. 9, 12)
- A. Geiger, Z. Wu, H. Lu, J. Rozner, E. Kreiss, T. Icard, N. D. Goodman, and C. Potts. Inducing causal structure for interpretable neural networks, 2022. URL <https://arxiv.org/abs/2112.00826>. (p. 12, 30)

- A. Geiger, C. Potts, and T. Icard. Causal abstraction for faithful model interpretation, 2023a. URL <https://arxiv.org/abs/2301.04709>. (p. 12, 30)
- A. Geiger, Z. Wu, C. Potts, T. Icard, and N. D. Goodman. Finding alignments between interpretable causal variables and distributed neural representations, 2023b. URL <https://arxiv.org/abs/2303.02536>. (p. 10, 12)
- G. Gemma Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, P. Tafti, L. Hussenot, P. G. Sessa, A. Chowdhery, A. Roberts, A. Barua, A. Botev, A. Castro-Ros, A. Slone, A. Héliou, A. Tacchetti, A. Bulanov, A. Paterson, B. Tsai, B. Shahriari, C. L. Lan, C. A. Choquette-Choo, C. Crepy, D. Cer, D. Ippolito, D. Reid, E. Buchatskaya, E. Ni, E. Noland, G. Yan, G. Tucker, G.-C. Muraru, G. Rozhdestvenskiy, H. Michalewski, I. Tenney, I. Grishchenko, J. Austin, J. Keeling, J. Labanowski, J.-B. Lespiau, J. Stanway, J. Brennan, J. Chen, J. Ferret, J. Chiu, J. Mao-Jones, K. Lee, K. Yu, K. Millican, L. L. Sjoesund, L. Lee, L. Dixon, M. Reid, M. Mikula, M. Wirth, M. Sharman, N. Chi-naev, N. Thain, O. Bachem, O. Chang, O. Wahltinez, P. Bailey, P. Michel, P. Yotov, R. Chaabouni, R. Comanescu, R. Jana, R. Anil, R. McIlroy, R. Liu, R. Mullins, S. L. Smith, S. Borgeaud, S. Girgin, S. Douglas, S. Pandya, S. Shakeri, S. De, T. Klimenko, T. Hennigan, V. Feinberg, W. Stokowiec, Y. hui Chen, Z. Ahmed, Z. Gong, T. Warkentin, L. Peran, M. Giang, C. Farabet, O. Vinyals, J. Dean, K. Kavukcuoglu, D. Hassabis, Z. Ghahramani, D. Eck, J. Barral, F. Pereira, E. Collins, A. Joulin, N. Fiedel, E. Senter, A. Andreev, and K. Kenealy. Gemma: Open models based on gemini research and technology. *ArXiv*, 2024. URL <https://arxiv.org/abs/2403.08295>. (p. 17)
- M. Geva, R. Schuster, J. Berant, and O. Levy. Transformer feed-forward layers are key-value memories. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL <https://aclanthology.org/2021.emnlp-main.446>. (p. 4)
- M. Geva, A. Caciularu, G. Dar, P. Roit, S. Sadde, M. Shlain, B. Tamir, and Y. Goldberg. LM-debugger: An interactive tool for inspection and intervention in transformer-based language models. In W. Che and E. Shutova (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 12–21, Abu Dhabi, UAE, December 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-demos.2. URL <https://aclanthology.org/2022.emnlp-demos.2>. (p. 30)
- M. Geva, A. Caciularu, K. Wang, and Y. Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Y. Goldberg, Z. Kozareva, and Y. Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 30–45, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.3. URL <https://aclanthology.org/2022.emnlp-main.3>. (p. 8, 22)
- M. Geva, J. Bastings, K. Filippova, and A. Globerson. Dissecting recall of factual associations in autoregressive language models. In H. Bouamor, J. Pino, and K. Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.751. URL <https://aclanthology.org/2023.emnlp-main.751>. (p. 11, 25, 28)
- A. Ghandeharioun, A. Caciularu, A. Pearce, L. Dixon, and M. Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. *Arxiv*, 2024. URL <https://arxiv.org/abs/2401.06102v2>. (p. 17)
- N. Goldowsky-Dill, C. MacLeod, L. Sato, and A. Arora. Localizing model behavior with path patching. *Arxiv*, 2023. URL <https://arxiv.org/abs/2304.05969>. (p. 11, 12)
- R. Gould, E. Ong, G. Ogden, and A. Conmy. Successor heads: Recurring, interpretable attention heads in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=kvcv8KQsi>. (p. 20)
- D. Groeneveld, I. Beltagy, P. Walsh, A. Bhagia, R. Kinney, O. Tafjord, A. H. Jha, H. Ivison, I. Magnusson, Y. Wang, S. Arora, D. Atkinson, R. Authur, K. R. Chandu, A. Cohan, J. Dumas, Y. Elazar, Y. Gu, J. Hessel, T. Khot, W. Merrill, J. Morrison, N. Muennighoff, A. Naik, C. Nam, M. E. Peters, V. Pyatkin, A. Ravichander, D. Schwenk, S. Shah, W. Smith, E. Strubell, N. Subramani, M. Wortsman, P. Dasigi, N. Lambert, K. Richardson, L. Zettlemoyer, J. Dodge, K. Lo, L. Soldaini, N. A. Smith, and H. Hajishirzi. Olmo: Accelerating the science of language models, 2024. (p. 4)
- R. B. Grosse, J. Bae, C. Anil, N. Elhage, A. Tamkin, A. Tajdini, B. Steiner, D. Li, E. Durmus, E. Perez, E. Hubinger, K. Lukovsiute, K. Nguyen, N. Joseph, S. McCandlish, J. Kaplan, and S. Bowman. Studying large language model generalization with influence functions. *ArXiv*, abs/2308.03296, 2023. URL <https://api.semanticscholar.org/CorpusID:260682872>. (p. 8)
- A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023. (p. 28)
- J.-C. Gu, H. Xu, J.-Y. Ma, P. Lu, Z.-H. Ling, K. wei Chang, and N. Peng. Model editing can hurt general abilities of large language models. *ArXiv*, abs/2401.04700, 2024. URL <https://api.semanticscholar.org/CorpusID:266899568>. (p. 29)
- C. Guerner, A. Svete, T. Liu, A. Warstadt, and R. Cotterell. A geometric notion of causal probing, 2023. (p. 11)

- N. M. Guerreiro, P. Colombo, P. Piantanida, and A. Martins. Optimal transport for unsupervised hallucination detection in neural machine translation. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13766–13784, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.770. URL <https://aclanthology.org/2023.acl-long.770>. (p. 27)
- N. M. Guerreiro, E. Voita, and A. Martins. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In A. Vlachos and I. Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1059–1075, Dubrovnik, Croatia, May 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.75. URL <https://aclanthology.org/2023.eacl-main.75>. (p. 27)
- A. Gupta, G. Boleda, M. Baroni, and S. Padó. Distributional vectors encode referential attributes. In L. Márquez, C. Callison-Burch, and J. Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 12–21, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1002. URL <https://aclanthology.org/D15-1002>. (p. 13)
- A. Gupta, A. Rao, and G. K. Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. *ArXiv*, abs/2401.07453, 2024a. URL <https://api.semanticscholar.org/CorpusID:266999650>. (p. 29)
- A. Gupta, D. Sajani, and G. Anumanchipalli. A unified framework for model editing, 2024b. (p. 29)
- W. Gurnee. Sae reconstruction errors are (empirically) pathological. *AI Alignment Forum*, 2024. URL <https://www.alignmentforum.org/posts/rZPiuFxEsmxCDHe4B/sae-reconstruction-errors-are-empirically-pathological>. (p. 16)
- W. Gurnee and M. Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=jE8xbmvFin>. (p. 22)
- W. Gurnee, N. Nanda, M. Pauly, K. Harvey, D. Troitskii, and D. Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=JYs1R9IMJr>. (p. 15, 22, 23)
- W. Gurnee, T. Horsley, Z. C. Guo, T. R. Kheirkhah, Q. Sun, W. Hathaway, N. Nanda, and D. Bertsimas. Universal neurons in gpt2 language models, 2024. (p. 21, 22, 23, 24, 25)
- K. Guu, A. Webson, E. Pavlick, L. Dixon, I. Tenney, and T. Bolukbasi. Simfluence: Modeling the influence of individual training examples by simulating training runs. *Arxiv*, 2023. URL <https://arxiv.org/abs/2303.08114>. (p. 8)
- Z. Hammoudeh and D. Lowd. Training data influence analysis and estimation: A survey, 2022. (p. 8)
- X. Han, B. C. Wallace, and Y. Tsvetkov. Explaining black box predictions and unveiling data artifacts through influence functions. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5553–5563, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.492. URL <https://aclanthology.org/2020.acl-main.492>. (p. 8)
- Z. Han, C. Gao, J. Liu, J. Zhang, and S. Q. Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey, 2024. (p. 30)
- M. Hanna, O. Liu, and A. Variengien. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 76033–76060. Curran Associates, Inc., 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/efbba7719cc5172d175240f24be11280-Abstract-Conference.html. (p. 10, 11, 25, 26)
- M. Hanna, S. Pezzelle, and Y. Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms, 2024. URL <https://arxiv.org/abs/2403.17806>. (p. 12, 26)
- P. Hase, M. Bansal, B. Kim, and A. Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 17643–17668. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3927bbdcf0e8d1fa8aa23c26f358a281-Paper-Conference.pdf. (p. 29)
- A. Haviv, I. Cohen, J. Gidron, R. Schuster, Y. Goldberg, and M. Geva. Understanding transformer memorization recall through idioms. In A. Vlachos and I. Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 248–264, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.19. URL <https://aclanthology.org/2023.eacl-main.19>. (p. 28)
- Z. He, X. Ge, Q. Tang, T. Sun, Q. Cheng, and X. Qiu. Dictionary learning improves patch-free circuit discovery in mechanistic interpretability: A case study on othello-gpt. *ArXiv*, abs/2402.12201, 2024. URL <https://api.semanticscholar.org/CorpusID:267751496>. (p. 16)
- S. Heimersheim and J. Janiak. A circuit for python docstrings in a 4-layer attention-only transformer. *AI Alignment Forum*, 2023. URL <https://www.alignmentforum.org/posts/u6KXXmKFbXfWzoAXn/a-circuit-for-python-docstrings-in-a-4-layer-attention-only>. (p. 10, 11, 22, 25)
- S. Heimersheim and N. Nanda. How to use and interpret activation patching. *Arxiv*, 2024. URL <https://arxiv.org/abs/2404.15255>. (p. 10)

-
- S. Heimersheim and A. Turner. Residual stream norms grow exponentially over the forward pass. *AI Alignment Forum*, 2023. URL <https://www.alignmentforum.org/posts/8mizBCm3dyc432nK8/residual-stream-norms-grow-exponentially-over-the-forward>. (p. 23)
- R. Hendel, M. Geva, and A. Globerson. In-context learning creates task vectors. *Arxiv*, 2023. URL <https://arxiv.org/abs/2310.15916>. (p. 25)
- E. Hernandez, A. S. Sharma, T. Haklay, K. Meng, M. Wattenberg, J. Andreas, Y. Belinkov, and D. Bau. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=w7LU2s14kE>. (p. 28)
- J. Hewitt and P. Liang. Designing and interpreting probes with control tasks. In K. Inui, J. Jiang, V. Ng, and X. Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL <https://aclanthology.org/D19-1275>. (p. 13)
- J. Hewitt and C. D. Manning. A structural probe for finding syntax in word representations. In J. Burstein, C. Doran, and T. Solorio (eds.), *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)*, pp. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL <https://aclanthology.org/N19-1419>. (p. 14)
- J. Hewitt, J. Thickstun, C. Manning, and P. Liang. Backpack language models. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9103–9125, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.506. URL <https://aclanthology.org/2023.acl-long.506>. (p. 29)
- A. Himmi, G. Staerman, M. Picot, P. Colombo, and N. M. Guerreiro. Enhanced hallucination detection in neural machine translation through simple detector aggregation, 2024. (p. 27)
- J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, O. Vinyals, J. Rae, and L. Sifre. An empirical analysis of compute-optimal large language model training. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c1e2faff6f588870935f114ebe04a3e5-Abstract-Conference.html. (p. 2)
- A. Holtzman, P. West, V. Shwartz, Y. Choi, and L. Zettlemoyer. Surface form competition: Why the highest probability answer isn’t always right. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.564. URL <https://aclanthology.org/2021.emnlp-main.564>. (p. 8)
- B. Hoover, H. Strobelt, and S. Gehrmann. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In A. Celikyilmaz and T.-H. Wen (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 187–196, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.22. URL <https://aclanthology.org/2020.acl-demos.22>. (p. 30)
- P. M. Htut, J. Phang, S. Bordia, and S. R. Bowman. Do attention heads in bert track syntactic dependencies? *Arxiv*, 2019. URL <https://arxiv.org/abs/1911.12246>. (p. 19)
- J. Huang, A. Geiger, K. D’Oosterlinck, Z. Wu, and C. Potts. Rigorously assessing natural language explanations of neurons. In Y. Belinkov, S. Hao, J. Jumelet, N. Kim, A. McCarthy, and H. Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 317–331, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.24. URL <https://aclanthology.org/2023.blackboxnlp-1.24>. (p. 18)
- J. Huang, Z. Wu, C. Potts, M. Geva, and A. Geiger. Ravel: Evaluating interpretability methods on disentangling language model representations, 2024a. URL <https://arxiv.org/abs/2402.17700>. (p. 12, 15)
- Y. Huang, S. Hu, X. Han, Z. Liu, and M. Sun. Unified view of grokking, double descent and emergent abilities: A perspective from circuits competition, 2024b. (p. 26)
- N. Hudson, J. G. Pauloski, M. Baughman, A. Kamatar, M. Sakarvadia, L. Ward, R. Chard, A. Bauer, M. Levental, W. Wang, W. Engler, O. P. Skelly, B. Blaiszik, R. Stevens, K. Chard, and I. Foster. Trillion parameter ai serving infrastructure for scientific discovery: A survey and vision. *Arxiv*, 2024. URL <https://arxiv.org/abs/2402.03480>. (p. 31)
- D. Hupkes, S. Veldhoen, and W. Zuidema. Visualisation and ‘diagnostic classifiers’ reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61(1): 907–926, 2018. ISSN 1076-9757. (p. 14)
- S. Jain, R. Kirk, E. S. Lubana, R. P. Dick, H. Tanaka, T. Rocktäschel, E. Grefenstette, and D. Krueger. What happens when you fine-tuning your model? mechanistic analysis of procedurally generated tasks. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=A0HKeK14N1>. (p. 26)

-
- S. Jain and B. C. Wallace. Attention is not Explanation. In J. Burstein, C. Doran, and T. Solorio (eds.), *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)*, pp. 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL <https://aclanthology.org/N19-1357>. (p. 7)
- S. Jastrzębski, D. Arpit, N. Ballas, V. Verma, T. Che, and Y. Bengio. Residual connections encourage iterative inference, 2018. (p. 17)
- A. Jermyn and A. Templeton. Circuits updates - jnauary 2024. ghost grads: An improvement on resampling. *Transformer Circuits Thread*, 2024. URL <https://transformer-circuits.pub/2024/jan-update/index.html>. (p. 55)
- Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), mar 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL <https://doi.org/10.1145/3571730>. (p. 26)
- Y. Jiang, G. Rajendran, P. Ravikumar, B. Aragam, and V. Veitch. On the origins of linear representations in large language models, 2024. (p. 14)
- S. Joseph. Vit prisma: A mechanistic interpretability library for vision transformers. *GitHub repository*, 2023. URL <https://github.com/soniajoseph/vit-prisma>. (p. 30)
- G. Kamradt. Needle in a haystack - pressure testing llms. *GitHub Repository*, 2023. URL https://github.com/gkamradt/LLMTest_NeedleInAHaystack. (p. 28)
- J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models, 2020. URL <https://arxiv.org/abs/2001.08361>. (p. 2)
- S. Katz and Y. Belinkov. VISIT: Visualizing and interpreting the semantic information flow of transformers. In H. Bouamor, J. Pino, and K. Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14094–14113, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.939. URL <https://aclanthology.org/2023.findings-emnlp.939>. (p. 30)
- S. Katz, Y. Belinkov, M. Geva, and L. Wolf. Backward lens: Projecting language model gradients into the vocabulary space, 2024. (p. 18)
- B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In J. Dy and A. Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2668–2677. PMLR, 2018. URL <https://proceedings.mlr.press/v80/kim18d.html>. (p. 13, 14)
- C. Kissane, R. Krzyzanowski, A. Conmy, and N. Nanda. Sparse autoencoders work on attention layer outputs. AI Alignment Forum, 2024a. URL <https://www.alignmentforum.org/posts/DtdzGwFh9dCfsekZZ>. (p. 22)
- C. Kissane, R. Krzyzanowski, A. Conmy, and N. Nanda. Attention saes scale to gpt-2 small. AI Alignment Forum, 2024b. URL <https://www.alignmentforum.org/posts/FSTRedtjuHa4Gfdr/attn-saes-scale-to-gpt-2-small>. (p. 16, 22)
- G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui. Attention is not only a weight: Analyzing transformers with vector norms. In B. Webber, T. Cohn, Y. He, and Y. Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.574. URL <https://aclanthology.org/2020.emnlp-main.574>. (p. 7, 21)
- G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui. Incorporating Residual and Normalization Layers into Analysis of Masked Language Models. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4547–4568, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.373. URL <https://aclanthology.org/2021.emnlp-main.373>. (p. 3, 7)
- G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui. Transformer language models handle word frequency in prediction head. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4523–4535, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.276. URL <https://aclanthology.org/2023.findings-acl.276>. (p. 23)
- G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui. Analyzing feed-forward blocks in transformers through the lens of attention map. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=mYwyTuiRp>. (p. 8)
- P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In D. Precup and Y. W. Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 06–11 Aug 2017. URL <https://proceedings.mlr.press/v70/koh17a.html>. (p. 8)

- A. Köhn. What's in an embedding? analyzing word embeddings through multilingual evaluation. In L. Márquez, C. Callison-Burch, and J. Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2067–2073, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1246. URL <https://aclanthology.org/D15-1246>. (p. 13)
- N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch. *Arxiv*, 2020. URL <https://arxiv.org/abs/2009.07896>. (p. 29)
- O. Kovaleva, A. Romanov, A. Rogers, and A. Rumshisky. Revealing the dark secrets of BERT. In K. Inui, J. Jiang, V. Ng, and X. Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4365–4374, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1445. URL <https://aclanthology.org/D19-1445>. (p. 19, 21)
- O. Kovaleva, S. Kulshreshtha, A. Rogers, and A. Rumshisky. BERT busters: Outlier dimensions that disrupt transformers. In C. Zong, F. Xia, W. Li, and R. Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3392–3405, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.300. URL <https://aclanthology.org/2021.findings-acl.300>. (p. 24)
- J. Kramár, T. Lieberum, R. Shah, and N. Nanda. Atp*: An efficient and scalable method for localizing llm behaviour to components, 2024. URL <https://arxiv.org/abs/2403.00745>. (p. 10, 12)
- R. Krzyzanowski, C. Kissane, A. Conmy, and N. Nanda. We inspected every head in GPT-2 small using saes so you don't have to. *AI Alignment Forum*, 2024. URL <https://www.alignmentforum.org/posts/xmegeW5mqiBsvoaIm/we-inspected-every-head-in-gpt-2-small-using-saes-so-you-don>. (p. 22)
- Y. Kwon, E. Wu, K. Wu, and J. Zou. Datainf: Efficiently estimating data influence in loRA-tuned LLMs and diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=9m02ib92Wz>. (p. 8)
- V. Lal, A. Ma, E. Aflalo, P. Howard, A. Simoes, D. Korat, O. Pereg, G. Singer, and M. Wasserblat. InterpreT: An interactive visualization tool for interpreting transformers. In D. Gkatzia and D. Seddah (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 135–142, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-demos.17. URL <https://aclanthology.org/2021.eacl-demos.17>. (p. 30)
- A. Langedijk, H. Mohebbi, G. Sarti, W. Zuidema, and J. Jumelet. Decoderlens: Layerwise interpretation of encoder-decoder transformers. *ArXiv*, abs/2310.03686, 2023. URL <https://api.semanticscholar.org/CorpusID:263671583>. (p. 17)
- T. Lanham, A. Chen, A. Radhakrishnan, B. Steiner, C. E. Denison, D. Hernandez, D. Li, E. Durmus, E. Hubinger, J. Kernion, K. Lukovsiute, K. Nguyen, N. Cheng, N. Joseph, N. Schiefer, O. Rausch, R. Larson, S. McCandlish, S. Kundu, S. Kadavath, S. Yang, T. Henighan, T. D. Maxwell, T. Telleen-Lawton, T. Hume, Z. Hatfield-Dodds, J. Kaplan, J. Brauner, S. Bowman, and E. Perez. Measuring faithfulness in chain-of-thought reasoning. *ArXiv*, abs/2307.13702, 2023. URL <https://api.semanticscholar.org/CorpusID:259953372>. (p. 19)
- K. Leino, S. Sen, A. Datta, M. Fredrikson, and L. Li. Influence-directed explanations for deep convolutional networks. In *2018 IEEE International Test Conference (ITC)*, pp. 1–8, 2018. doi: 10.1109/TEST.2018.8624792. (p. 12)
- M. A. Lepori, T. Serre, and E. Pavlick. Uncovering intermediate variables in transformers using circuit probing, 2023. (p. 12)
- J. Li, X. Chen, E. Hovy, and D. Jurafsky. Visualizing and understanding neural models in NLP. In K. Knight, A. Nenkova, and O. Rambow (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 681–691, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1082. URL <https://aclanthology.org/N16-1082>. (p. 6)
- J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure, 2017. (p. 7)
- K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL <https://openreview.net/forum?id=aLLuYpn83y>. (p. 26, 27)
- X. L. Li, A. Holtzman, D. Fried, P. Liang, J. Eisner, T. Hashimoto, L. Zettlemoyer, and M. Lewis. Contrastive decoding: Open-ended text generation as optimization. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12286–12312, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.687. URL <https://aclanthology.org/2023.acl-long.687>. (p. 29)
- Z. Li, N. Zhang, Y. Yao, M. Wang, X. Chen, and H. Chen. Unveiling the pitfalls of knowledge editing for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=fNktD3ib16>. (p. 29)
- Q. V. Liao, D. Gruen, and S. Miller. Questioning the ai: Informing design practices for explainable ai user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pp. 1–15, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376590. URL <https://doi.org/10.1145/3313831.3376590>. (p. 31)

-
- T. Lieberum, M. Rahtz, J. Kramár, N. Nanda, G. Irving, R. Shah, and V. Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *Arxiv*, 2023. URL <https://arxiv.org/abs/2307.09458>. (p. 10, 12)
- J. Lin and J. Bloom. Announcing neuronpedia: Platform for accelerating research into sparse autoencoders. *AI Alignment Forum*, 2024. URL <https://www.alignmentforum.org/posts/BaEQoxHhWPrkinmxd/announcing-neuronpedia-platform-for-accelerating-research>. (p. 16, 30)
- Y. Lin, Y. C. Tan, and R. Frank. Open sesame: Getting inside BERT’s linguistic knowledge. In T. Linzen, G. Chrupała, Y. Belinkov, and D. Hupkes (eds.), *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 241–253, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4825. URL <https://aclanthology.org/W19-4825>. (p. 14)
- D. Lindner, J. Kramar, S. Farquhar, M. Rahtz, T. McGrath, and V. Mikulik. Tracr: Compiled transformers as a laboratory for interpretability. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 37876–37899. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/771155abaee744e08576f1f3b4b7ac0d-Paper-Conference.pdf. (p. 30)
- Z. C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, jun 2018. ISSN 1542-7730. doi: 10.1145/3236386.3241340. URL <https://doi.org/10.1145/3236386.3241340>. (p. 6)
- N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith. Linguistic knowledge and transferability of contextual representations. In J. Burstein, C. Doran, and T. Solorio (eds.), *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)*, pp. 1073–1094, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1112. URL <https://aclanthology.org/N19-1112>. (p. 14)
- Q. Liu, Y. Chai, S. Wang, Y. Sun, K. Wang, and H. Wu. On training data influence of gpt models. *Arxiv*, 2024. URL <https://arxiv.org/abs/2404.07840>. (p. 8)
- L. Longo, M. Bricé, F. Cabitza, J. Choi, R. Confalonieri, J. D. Ser, R. Guidotti, Y. Hayashi, F. Herrera, A. Holzinger, R. Jiang, H. Khosravi, F. Lecue, G. Malgieri, A. Pérez, W. Samek, J. Schneider, T. Speith, and S. Stumpf. Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, 106:102301, 2024. ISSN 1566-2535. doi: <https://doi.org/10.1016/j.inffus.2024.102301>. URL <https://www.sciencedirect.com/science/article/pii/S1566253524000794>. (p. 31)
- M. Loog, T. Viering, A. Mey, J. H. Krijthe, and D. M. J. Tax. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20):10625–10626, 2020. doi: 10.1073/pnas.2001875117. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2001875117>. (p. 26)
- S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf. (p. 7, 29)
- Z. Luo, A. Kulmizev, and X. Mao. Positional artefacts propagate through masked language model embeddings. In C. Zong, F. Xia, W. Li, and R. Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5312–5327, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.413. URL <https://aclanthology.org/2021.acl-long.413>. (p. 24)
- A. Lv, K. Zhang, Y. Chen, Y. Wang, L. Liu, J.-R. Wen, J. Xie, and R. Yan. Interpreting key mechanisms of factual recall in transformer-based language models. *Computing Research Repository*, arXiv:2403.19521, 2024. URL <https://arxiv.org/abs/2403.19521>. (p. 21, 28)
- Q. Lyu, M. Apidianaki, and C. Callison-Burch. Towards Faithful Model Explanation in NLP: A Survey. *Computational Linguistics*, pp. 1–70, 01 2024. ISSN 0891-2017. doi: 10.1162/coli_a_00511. URL https://doi.org/10.1162/coli_a_00511. (p. 2)
- M. MacDiarmid, T. Maxwell, N. Schiefer, J. Mu, J. Kaplan, D. Duvenaud, S. Bowman, A. Tamkin, E. Perez, M. Sharma, C. Denison, and E. Hubinger. Simple probes can catch sleeper agents. *Anthropic*, 2024. URL <https://www.anthropic.com/news/probes-catch-sleeper-agents>. (p. 14)
- A. Madsen, S. Reddy, and S. Chandar. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8), 2022. ISSN 0360-0300. doi: 10.1145/3546577. URL <https://doi.org/10.1145/3546577>. (p. 2, 6)
- A. Madsen, S. Chandar, and S. Reddy. Are self-explanations from large language models faithful? *ArXiv*, abs/2401.07927, 2024. URL <https://api.semanticscholar.org/CorpusID:266999774>. (p. 19)
- A. Makelov, G. Lange, A. Geiger, and N. Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=Ebt7JgMHv1>. (p. 12)
- S. Marks and A. Mueller. Dictionary learning. *GitHub repository*, 2023. URL https://github.com/saprmarks/dictionary_learning. (p. 30)
- S. Marks and M. Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets, 2023. URL <https://arxiv.org/abs/2310.06824>. (p. 15)

-
- S. Marks, C. Rager, E. J. Michaud, Y. Belinkov, D. Bau, and A. Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *Computing Research Repository*, arXiv:2403.19647, 2024. URL <https://arxiv.org/abs/2403.19647>. (p. 12, 16, 31)
- T. McCoy, E. Pavlick, and T. Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In A. Korhonen, D. Traum, and L. Márquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL <https://aclanthology.org/P19-1334>. (p. 8)
- C. McDougall. Six (and a half) intuitions for SVD. *Callum McDougall Blog*, 2023. URL <https://www.perfectlynormal.co.uk/blog-kl-divergence>. (p. 18)
- C. McDougall and J. Bloom. Sae-vis: Announcement post. *LessWrong*, 2024. URL <https://www.lesswrong.com/posts/nAhy6ZquNY7AD3RkD/sae-vis-announcement-post-1>. (p. 30)
- C. McDougall, A. Conmy, C. Rushing, T. McGrath, and N. Nanda. Copy suppression: Comprehensively understanding an attention head. *Arxiv*, 2023. URL <https://arxiv.org/abs/2310.04625>. (p. 20)
- T. McGrath, A. Kapishnikov, N. Tomašev, A. Pearce, M. Wattenberg, D. Hassabis, B. Kim, U. Paquet, and V. Kramnik. Acquisition of chess knowledge in alphazero. *Proceedings of the National Academy of Sciences*, 119(47):e2206625119, 2022. doi: 10.1073/pnas.2206625119. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2206625119>. (p. 13)
- T. McGrath, M. Rahtz, J. Kramar, V. Mikulik, and S. Legg. The hydra effect: Emergent self-repair in language model computations. *Arxiv*, 2023. URL <https://arxiv.org/abs/2307.15771>. (p. 9, 12)
- K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in GPT. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6fd43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html. (p. 9, 10, 28, 29)
- K. Meng, A. S. Sharma, A. J. Andonian, Y. Belinkov, and D. Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=MkbcAHlYgyS>. (p. 29)
- W. Merrill, V. Ramanujan, Y. Goldberg, R. Schwartz, and N. A. Smith. Effects of parameter norm growth during transformer training: Inductive bias from gradient descent. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1766–1781, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.133. URL <https://aclanthology.org/2021.emnlp-main.133>. (p. 23)
- W. Merrill, N. Tsilivis, and A. Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. *ArXiv*, abs/2303.11873, 2023. URL <https://api.semanticscholar.org/CorpusID:257636667>. (p. 26)
- J. Merullo, C. Eickhoff, and E. Pavlick. A mechanism for solving relational tasks in transformer language models, 2023. URL <https://arxiv.org/abs/2305.16130>. (p. 28)
- J. Merullo, C. Eickhoff, and E. Pavlick. Circuit component reuse across tasks in transformer language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=fpoAYV6Wsk>. (p. 26)
- P. Michel, O. Levy, and G. Neubig. Are sixteen heads really better than one? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2c601ad9d2ff9bc8b282670cdd54f69f-Paper.pdf. (p. 8, 10)
- T. Mickus, D. Paperno, and M. Constant. How to dissect a Muppet: The structure of transformer embedding spaces. *Transactions of the Association for Computational Linguistics*, 10:981–996, 2022. doi: 10.1162/tac1_a_00501. URL <https://aclanthology.org/2022.tacl-1.57>. (p. 5)
- V. Miglani, A. Yang, A. Markosyan, D. Garcia-Olano, and N. Kokhlikyan. Using captum to explain generative language models. In L. Tan, D. Milajevs, G. Chauhan, J. Gwinnup, and E. Rippeth (eds.), *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pp. 165–173, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.nlposs-1.19. URL <https://aclanthology.org/2023.nlposs-1.19>. (p. 29)
- T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html. (p. 14)
- B. Millidge and S. Black. The singular value decompositions of transformer weight matrices are highly interpretable. *AI Alignment Forum*, 2022. URL <https://www.alignmentforum.org/posts/mkbGjzx08d8XqKHZA/the-singular-value-decompositions-of-transformer-weight>. (p. 17)
- B. Millidge and E. Winsor. Basic facts about language model internals. *AI Alignment Forum*, 2023. URL <https://www.alignmentforum.org/posts/PDLfPwSynu73mxGw/basic-facts-about-language-model-internals-1>. (p. 23)

-
- S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao. Large language models: A survey, 2024. URL <https://arxiv.org/abs/2402.06196>. (p. 26)
- E. Mitchell, C. Lin, A. Bosselut, C. Finn, and C. D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a. URL <https://openreview.net/forum?id=0DcZxeWfOPt>. (p. 29)
- E. Mitchell, C. Lin, A. Bosselut, C. D. Manning, and C. Finn. Memory-based model editing at scale. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831. PMLR, 17–23 Jul 2022b. URL <https://proceedings.mlr.press/v162/mitchell122a.html>. (p. 29)
- A. Modarressi, M. Fayyaz, Y. Yaghoobzadeh, and M. T. Pilehvar. GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 258–271, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.19. URL <https://aclanthology.org/2022.naacl-main.19>. (p. 7)
- A. Modarressi, M. Fayyaz, E. Aghazadeh, Y. Yaghoobzadeh, and M. T. Pilehvar. DecompX: Explaining transformers decisions by propagating token decomposition. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2649–2664, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.149. URL <https://aclanthology.org/2023.acl-long.149>. (p. 8)
- H. Mohebbi, W. Zuidema, G. Chrupala, and A. Alishahi. Quantifying context mixing in transformers. In A. Vlachos and I. Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3378–3400, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.245. URL <https://aclanthology.org/2023.eacl-main.245>. (p. 7, 10)
- R. Molina. Traveling words: A geometric interpretation of transformers. *Arxiv*, 2023. URL <https://arxiv.org/abs/2309.07315>. (p. 18)
- G. Monea, M. Peyrard, M. Josifoski, V. Chaudhary, J. Eisner, E. Kıcıman, H. Palangi, B. Patra, and R. West. A glitch in the matrix? locating and detecting language model grounding with fakepedia, 2024. URL <https://arxiv.org/abs/2312.02073>. (p. 28)
- D. Mossing, S. Bills, H. Tillman, T. Dupré la Tour, N. Cammarata, L. Gao, J. Achiam, C. Yeh, J. Leike, J. Wu, and W. Saunders. Transformer debugger. <https://github.com/openai/transformer-debugger>, 2024. (p. 30)
- N. Nanda. Induction mosaic. *Neel Nanda Blog*, 2022a. URL <https://neelnanda.io/mosaic>. (p. 20)
- N. Nanda. Neuroscope: A website for mechanistic interpretability of language models. *Website*, 2022b. URL <https://neuroscope.io/>. (p. 18)
- N. Nanda. Attribution patching: Activation patching at industrial scale. <https://www.neelnanda.io/mechanistic-interpretability/attribution-patching>, 2023. (p. 12)
- N. Nanda and J. Bloom. Transformerlens. *Github Repository*, 2022. URL <https://github.com/neelnanda-io/TransformerLens>. (p. 29)
- N. Nanda, L. Chan, T. Lieberum, J. Smith, and J. Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023a. URL <https://openreview.net/forum?id=9XFSbDPmdW>. (p. 26)
- N. Nanda, A. Lee, and M. Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Y. Belinkov, S. Hao, J. Jumelet, N. Kim, A. McCarthy, and H. Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2. URL <https://aclanthology.org/2023.blackboxnlp-1.2>. (p. 14)
- N. Nanda, S. Rajamanoharan, J. Kramár, and R. Shah. Fact finding: Attempting to reverse-engineer factual recall on the neuron level. *AI Alignment Forum*, 2023c. URL <https://www.alignmentforum.org/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factual-recall>. (p. 19, 28)
- C. Neo, S. B. Cohen, and F. Barez. Interpreting context look-ups in transformers: Investigating attention-mlp interactions, 2024. URL <https://arxiv.org/abs/2402.15055>. (p. 25)
- A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16*, pp. 3395–3403, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819. (p. 18)
- R. Nogueira, Z. Jiang, and J. Lin. Investigating the limitations of transformers with simple arithmetic tasks, 2021. (p. 31)
- nostalgebraist. Interpreting GPT: the logit lens. *AI Alignment Forum*, 2020. URL <https://www.alignmentforum.org/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens>. (p. 17)

- B.-D. Oh and W. Schuler. Token-wise decomposition of autoregressive language model hidden states for analyzing model predictions. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10105–10117, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.562. URL <https://aclanthology.org/2023.acl-long.562>. (p. 8)
- C. Olah. Mechanistic interpretability, variables, and the importance of interpretable bases. *Transformer Circuits Thread*, 2022. URL <https://transformer-circuits.pub/2022/mech-interp-essay>. (p. 11, 13)
- C. Olah. Distributed representations: Composition & superposition. *Transformer Circuits Thread*, 2023. URL <https://transformer-circuits.pub/2023/superposition-composition/index.html>. (p. 15)
- C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. An overview of early vision in inceptionv1. *Distill*, 2020a. doi: 10.23915/distill.00024.002. <https://distill.pub/2020/circuits/early-vision>. (p. 23)
- C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. Zoom in: An introduction to circuits. *Distill*, 2020b. doi: 10.23915/distill.00024.001. URL <https://distill.pub/2020/circuits/zoom-in>. (p. 15, 23)
- B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997. ISSN 0042-6989. doi: [https://doi.org/10.1016/S0042-6989\(97\)00169-7](https://doi.org/10.1016/S0042-6989(97)00169-7). URL <https://www.sciencedirect.com/science/article/pii/S0042698997001697>. (p. 15)
- C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. URL <https://transformer-circuit-s.pub/2022/in-context-learning-and-induction-heads/index.html>. (p. 10, 19, 20, 26)
- F. Ortu, Z. Jin, D. Doimo, M. Sachan, A. Cazzaniga, and B. Schölkopf. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. *Computing Research Repository*, arXiv:2402.11655, 2024. URL <https://arxiv.org/abs/2402.11655>. (p. 28)
- G. PAIR Team. Saliency: Framework-agnostic implementation for state-of-the-art saliency methods, 2023. URL <https://github.com/PAIR-code/saliency>. (p. 29)
- K. Pal, J. Sun, A. Yuan, B. Wallace, and D. Bau. Future lens: Anticipating subsequent tokens from a single hidden state. In J. Jiang, D. Reitter, and S. Deng (eds.), *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pp. 548–560, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.conll-1.37. URL <https://aclanthology.org/2023.conll-1.37>. (p. 17)
- L. Parcalabescu and A. Frank. On measuring faithfulness or self-consistency of natural language explanations, 2023. (p. 19)
- K. Park, Y. J. Choe, and V. Veitch. The linear representation hypothesis and the geometry of large language models. *Arxiv*, 2023a. URL <https://arxiv.org/abs/2311.03658>. (p. 14)
- S. M. Park, K. Georgiev, A. Ilyas, G. Leclerc, and A. Mądry. Trak: attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org, 2023b. (p. 8)
- G. Paulo, T. Marshall, and N. Belrose. Does transformer interpretability transfer to rnns?, 2024. (p. 15)
- J. Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI’01, pp. 411–420, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608001. (p. 12)
- J. Pearl. *Causality*. Cambridge University Press, 2 edition, 2009. doi: 10.1017/CBO9780511803161. (p. 9)
- M. E. Peters, M. Neumann, L. Zettlemoyer, and W.-t. Yih. Dissecting contextual word embeddings: Architecture and representation. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1179. URL <https://aclanthology.org/D18-1179>. (p. 13)
- P. Pezeshkpour, S. Jain, S. Singh, and B. Wallace. Combining feature and instance attribution to detect artifacts. In S. Muresan, P. Nakov, and A. Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1934–1946, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.153. URL <https://aclanthology.org/2022.findings-acl.153>. (p. 8)
- C. Pierse. Transformers Interpret, February 2021. URL <https://github.com/cdpierse/transformers-interpret>. (p. 29)
- T. Pimentel, J. Valvoda, R. H. Maudslay, R. Zmigrod, A. Williams, and R. Cotterell. Information-theoretic probing for linguistic structure. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4609–4622, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.420. URL <https://aclanthology.org/2020.acl-main.420>. (p. 14)
- A. Power, Y. Burda, H. Edwards, I. Babuschkin, and V. Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. (p. 26)

-
- N. Prakash, T. R. Shaham, T. Haklay, Y. Belinkov, and D. Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=8sKcAWOf2D>. (p. 26)
- G. Puccetti, A. Rogers, A. Drozd, and F. Dell’Orletta. Outlier dimensions that disrupt transformers are driven by frequency. In Y. Goldberg, Z. Kozareva, and Y. Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1286–1304, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.93. URL <https://aclanthology.org/2022.findings-emnlp.93>. (p. 21, 24)
- J. Qi, R. Fernández, and A. Bisazza. Cross-lingual consistency of factual knowledge in multilingual language models. In H. Bouamor, J. Pino, and K. Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10650–10666, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.658. URL <https://aclanthology.org/2023.emnlp-main.658>. (p. 29)
- L. Quirke, L. Heindrich, W. Gurnee, and N. Nanda. Training dynamics of contextual n-grams in language models, 2023. URL <https://arxiv.org/abs/2311.00863>. (p. 22)
- A. Radford, R. Jozefowicz, and I. Sutskever. Learning to generate reviews and discovering sentiment. *Arxiv*, 2017. URL <https://arxiv.org/abs/1704.01444>. (p. 14)
- A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018. URL <https://openai.com/research/language-unsupervised>. (p. 2)
- A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019. URL https://d4mucfpksyv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. (p. 2, 21)
- A. Raganato and J. Tiedemann. An analysis of encoder representations in transformer-based machine translation. In T. Linzen, G. Chrupala, and A. Alishahi (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 287–297, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5431. URL <https://aclanthology.org/W18-5431>. (p. 19)
- S. Rajamanoharan. Progress update 1 from the gdm mech interp team. improving ghost grads. *AI Alignment Forum*, 2024. URL <https://www.alignmentforum.org/posts/C5KAZQib3bzzpeyrg/progress-update-1-from-the-gdm-mech-interp-team-full-update>. (p. 55)
- S. Rajamanoharan, A. Conmy, L. Smith, T. Lieberum, V. Varma, J. Kramár, R. Shah, and N. Nanda. Improving dictionary learning with gated sparse autoencoders. *ArXiv*, 2024. (p. 16, 17, 55)
- S. Ravfogel, Y. Elazar, H. Gonen, M. Twiton, and Y. Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7237–7256, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.647. URL <https://aclanthology.org/2020.acl-main.647>. (p. 14)
- S. Ravfogel, M. Twiton, Y. Goldberg, and R. D. Cotterell. Linear adversarial concept erasure. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18400–18421. PMLR, 17–23 Jul 2022. URL <https://proceedings.mlr.press/v162/ravfogel22a.html>. (p. 14)
- M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016. (p. 7)
- A. Rogers, O. Kovaleva, and A. Rumshisky. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 01 2021. ISSN 2307-387X. doi: 10.1162/tac1_a_00349. URL https://doi.org/10.1162/tac1_a_00349. (p. 2, 14)
- W. Rudman, C. Chen, and C. Eickhoff. Outlier dimensions encode task specific knowledge. In H. Bouamor, J. Pino, and K. Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14596–14605, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.901. URL <https://aclanthology.org/2023.emnlp-main.901>. (p. 24)
- C. Rushing and N. Nanda. Explorations of self-repair in language models, 2024. URL <https://arxiv.org/abs/2402.15390>. (p. 12)
- T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. *Arxiv*, 2023. URL <https://arxiv.org/abs/2207.13243>. (p. 2)
- M. Sakarvadia, A. Khan, A. Ajith, D. Grzenda, N. Hudson, A. Bauer, K. Chard, and I. Foster. Attention lens: A tool for mechanistically interpreting the attention head information retrieval mechanism, 2023. (p. 17)
- S. Sanyal and X. Ren. Discretized integrated gradients for explaining language models. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10285–10299, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.805. URL <https://aclanthology.org/2021.emnlp-main.805>. (p. 7)

- G. Sarti, N. Feldhus, L. Sickert, O. van der Wal, M. Nissim, and A. Bisazza. Inseq: An interpretability toolkit for sequence generation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 421–435, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-demo.40. URL <https://aclanthology.org/2023.acl-demo.40>. (p. 12, 29)
- G. Sarti, G. Chrupała, M. Nissim, and A. Bisazza. Quantifying the plausibility of context reliance in neural machine translation. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria, May 2024. OpenReview. URL <https://openreview.net/forum?id=XTHfNGI3zT>. (p. 29)
- T. R. Shaham, S. Schwettmann, F. Wang, A. Rajaram, E. Hernandez, J. Andreas, and A. Torralba. A multimodal automated interpretability agent. *Arxiv*, 2024. URL <https://arxiv.org/abs/2404.14394>. (p. 30)
- L. S. Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, Princeton, 1953. (p. 7)
- L. Sharkey, D. Braun, and B. Millidge. Taking features out of superposition with sparse autoencoders. *AI Alignment Forum*, 2022. URL <https://www.alignmentforum.org/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition>. (p. 15)
- A. S. Sharma, D. Atkinson, and D. Bau. Locating and editing factual associations in mamba, 2024a. (p. 28)
- P. Sharma, J. T. Ash, and D. Misra. The truth is in there: Improving reasoning with layer-selective rank reduction. In *The Twelfth International Conference on Learning Representations*, 2024b. URL <https://openreview.net/forum?id=ozX92bu8VA>. (p. 18, 28)
- N. Shazeer. Glue variants improve transformer. *ArXiv*, 2020. (p. 16)
- A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML’17*, pp. 3145–3153. JMLR.org, 2017. (p. 6)
- A. Shrikumar, J. Su, and A. Kundaje. Computationally efficient measures of internal neuron importance. *ArXiv*, abs/1807.09946, 2018. URL <https://api.semanticscholar.org/CorpusID:50787065>. (p. 12)
- N. Y. Siegel, O.-M. Camburu, N. Heess, and M. Perez-Ortiz. The probabilities also matter: A more faithful metric for faithfulness of free-text explanations in large language models, 2024. (p. 31)
- K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *The Second International Conference on Learning Representations*, 2014. URL <http://arxiv.org/abs/1312.6034>. (p. 6, 18)
- A. K. Singh, T. Moskovitz, F. Hill, S. C. Y. Chan, and A. M. Saxe. What needs to go right for an induction head? a mechanistic study of in-context learning circuits and their formation, 2024a. (p. 20)
- C. Singh, J. P. Inala, M. Galley, R. Caruana, and J. Gao. Rethinking interpretability in the era of large language models. *ArXiv*, abs/2402.01761, 2024b. URL <https://api.semanticscholar.org/CorpusID:267412530>. (p. 31)
- S. Singh, S. Ravfogel, J. Herzig, R. Aharoni, R. Cotterell, and P. Kumaraguru. Mimic: Minimally modified counterfactuals in the representation space. *Arxiv*, 2024c. URL <https://arxiv.org/abs/2402.09631>. (p. 14, 15)
- L. Sixt, M. Granz, and T. Landgraf. When explanations lie: Why many modified BP attributions fail. In H. D. III and A. Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9046–9057. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/sixt20a.html>. (p. 8)
- D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise, 2017. (p. 6)
- P. Smolensky. *Neural and conceptual interpretation of PDP models*, pp. 390–431. MIT Press, Cambridge, MA, USA, 1986. ISBN 0262631105. (p. 15)
- N. Stoeck, M. Gordon, C. Zhang, and O. Lewis. Localizing paragraph memorization in language models, 2024. URL <https://arxiv.org/abs/2403.19851>. (p. 28)
- A. Stolfo, Y. Belinkov, and M. Sachan. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In H. Bouamor, J. Pino, and K. Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7035–7052, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.435. URL <https://aclanthology.org/2023.emnlp-main.435>. (p. 11)
- A. Stolfo, Y. Belinkov, and M. Sachan. Understanding arithmetic reasoning in language models using causal mediation analysis. *Arxiv*, 2023b. URL <https://arxiv.org/abs/2305.15054>. (p. 11, 25)
- X. Suau, L. Zappella, and N. Apostoloff. Finding experts in transformer models, 2020. (p. 22)
- X. Suau, L. Zappella, and N. Apostoloff. Self-conditioning pre-trained language models. *International Conference on Machine Learning*, 2022. URL <https://proceedings.mlr.press/v162/cuadros22a/cuadros22a.pdf>. (p. 22)
- M. Sun, X. Chen, J. Z. Kolter, and Z. Liu. Massive activations in large language models, 2024. URL <https://arxiv.org/abs/2402.17762>. (p. 21)
- M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3319–3328. JMLR.org, 2017. (p. 6)

-
- A. Syed, C. Rager, and A. Conmy. Attribution patching outperforms automated circuit discovery. *Arxiv*, 2023. URL <https://arxiv.org/abs/2310.10348>. (p. 12)
- S. Takase, S. Kiyono, S. Kobayashi, and J. Suzuki. B2T connection: Serving stability and performance in deep transformers. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 3078–3095, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.192. URL <https://aclanthology.org/2023.findings-acl.192>. (p. 3)
- T. Tang, W. Luo, H. Huang, D. Zhang, X. Wang, X. Zhao, F. Wei, and J.-R. Wen. Language-specific neurons: The key to multilingual capabilities in large language models, 2024. URL <https://arxiv.org/abs/2402.16438>. (p. 22)
- D. A. Tarzanagh, Y. Li, C. Thrampoulidis, and S. Oymak. Transformers as support vector machines, 2024. (p. 31)
- A. Templeton, T. Conerly, J. Marcus, T. Henighan, A. Golubeva, and T. Bricken. Circuits updates - february 2024. update on dictionary learning improvements. *Transformer Circuits Thread*, 2024. URL <https://transformer-circuits.pub/2024/feb-update/index.html>. (p. 55)
- I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. In A. Korhonen, D. Traum, and L. Márquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL <https://aclanthology.org/P19-1452>. (p. 14)
- I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. V. Durme, S. Bowman, D. Das, and E. Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019b. URL <https://openreview.net/forum?id=SJzSgnRcKX>. (p. 14)
- I. Tenney, J. Wexler, J. Bastings, T. Bolukbasi, A. Coenen, S. Gehrmann, E. Jiang, M. Pushkarna, C. Radebaugh, E. Reif, and A. Yuan. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. In Q. Liu and D. Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 107–118, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.15. URL <https://aclanthology.org/2020.emnlp-demos.15>. (p. 29)
- I. Tenney, R. Mullins, B. Du, S. Pandya, M. Kahng, and L. Dixon. Interactive prompt debugging with sequence salience. *Arxiv*, 2024. URL <https://arxiv.org/abs/2404.07498>. (p. 29)
- Y. Tian, Y. Wang, B. Chen, and S. Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer, 2023. (p. 31)
- Y. Tian, Y. Wang, Z. Zhang, B. Chen, and S. S. Du. JoMA: Demystifying multilayer transformers via joint dynamics of MLP and attention. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=LbJqRGNYCf>. (p. 31)
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. doi: <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>. URL <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1996.tb02080.x>. (p. 16)
- C. Tigges, O. J. Hollinsworth, A. Geiger, and N. Nanda. Linear representations of sentiment in large language models. *Arxiv*, 2023. URL <https://arxiv.org/abs/2310.15154>. (p. 14, 15, 24)
- W. Timkey and M. van Schijndel. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4527–4546, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.372. URL <https://aclanthology.org/2021.emnlp-main.372>. (p. 24)
- E. Todd, M. Li, A. S. Sharma, A. Mueller, B. C. Wallace, and D. Bau. LLMs represent contextual tasks as compact function vectors. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=AwxytyMwaG>. (p. 25)
- H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models. *Arxiv*, 2023. URL <https://arxiv.org/abs/2307.09288>. (p. 3, 4, 21)
- I. Tufanov, K. Hambardzumyan, J. Ferrando, and E. Voita. Lm transparency tool: Interactive tool for analyzing transformer language models. *Arxiv*, 2024. URL <https://arxiv.org/abs/2404.07004>. (p. 30)
- A. M. Turner, L. Thiergart, D. Udell, G. Leech, U. Mini, and M. MacDiarmid. Activation addition: Steering language models without optimization, 2023. (p. 14)
- M. Turpin, J. Michael, E. Perez, and S. Bowman. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *ArXiv*, abs/2305.04388, 2023. URL <https://api.semanticscholar.org/CorpusID:258556812>. (p. 19)

- A. Variengien. Some common confusion about induction heads. *LessWrong*, 2023. URL <https://www.lesswrong.com/posts/nJqftacoQGkurJ6fv/some-common-confusion-about-induction-heads>. (p. 20)
- A. Variengien and E. Winsor. Look before you leap: A universal emergent decomposition of retrieval tasks in language models, 2023. URL <https://arxiv.org/abs/2312.10091>. (p. 25, 28)
- V. Varma, R. Shah, Z. Kenton, J. Kram’ar, and R. Kumar. Explaining grokking through circuit efficiency. *ArXiv*, abs/2309.02390, 2023. URL <https://api.semanticscholar.org/CorpusID:261557247>. (p. 26)
- N. Varshney, W. Yao, H. Zhang, J. Chen, and D. Yu. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation, 2023. URL <https://arxiv.org/abs/2307.03987>. (p. 26)
- H. Vasconcelos, M. Jörke, M. Grunde-McLaughlin, T. Gerstenberg, M. S. Bernstein, and R. Krishna. Explanations can reduce overreliance on ai systems during decision-making. *Proc. ACM Hum.-Comput. Interact.*, 7 (CSCW1), apr 2023. doi: 10.1145/3579605. URL <https://doi.org/10.1145/3579605>. (p. 31)
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>. (p. 2)
- A. Veit, M. Wilber, and S. Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16*, pp. 550–558, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819. (p. 5)
- J. Vig. A multiscale visualization of attention in the transformer model. In M. R. Costa-jussà and E. Alfonseca (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3007. URL <https://aclanthology.org/P19-3007>. (p. 30)
- J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, Y. Singer, and S. Shieber. Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html>. (p. 9, 12)
- E. Voita and I. Titov. Information-theoretic probing with minimum description length. In B. Webber, T. Cohn, Y. He, and Y. Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 183–196, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.14. URL <https://aclanthology.org/2020.emnlp-main.14>. (p. 14)
- E. Voita, R. Sennrich, and I. Titov. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In K. Inui, J. Jiang, V. Ng, and X. Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4396–4406, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1448. URL <https://aclanthology.org/D19-1448>. (p. 31)
- E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In A. Korhonen, D. Traum, and L. Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL <https://aclanthology.org/P19-1580>. (p. 8, 10, 19)
- E. Voita, R. Sennrich, and I. Titov. Analyzing the source and target contributions to predictions in neural machine translation. In C. Zong, F. Xia, W. Li, and R. Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1126–1140, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.91. URL <https://aclanthology.org/2021.acl-long.91>. (p. 7)
- E. Voita, J. Ferrando, and C. Nalmpantis. Neurons in large language models: Dead, n-gram, positional. *Arxiv*, 2023. URL <https://arxiv.org/abs/2309.04827>. (p. 4, 14, 22, 23, 25)
- J. Von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov. Transformers learn in-context by gradient descent. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/von-oswald23a.html>. (p. 31)
- K. R. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023a. URL <https://openreview.net/forum?id=NpsVSN6o4u1>. (p. 9, 10, 11, 19, 25, 26)
- Q. Wang, T. Anikina, N. Feldhus, J. van Genabith, L. Hennig, and S. Möller. Llmcheckup: Conversational examination of large language models via interpretability tools, 2024. (p. 31)
- S. Wang, Y. Zhu, H. Liu, Z. Zheng, C. Chen, and J. Li. Knowledge editing for large language models: A survey. *ArXiv*, abs/2310.16218, 2023b. URL <https://api.semanticscholar.org/CorpusID:264487359>. (p. 29)

-
- X. Wang, K. Wen, Z. Zhang, L. Hou, Z. Liu, and J. Li. Finding skill neurons in pre-trained transformer-based language models. In Y. Goldberg, Z. Kozareva, and Y. Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11132–11152, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.765. URL <https://aclanthology.org/2022.emnlp-main.765>. (p. 22)
- D. Wei, R. Nair, A. Dhurandhar, K. R. Varshney, E. Daly, and M. Singh. On the safety of interpretable machine learning: A maximum deviation approach. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 9866–9880. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/402e12102d6ec3ea3df40ce1b23d423a-Paper-Conference.pdf. (p. 2)
- G. Weiss, Y. Goldberg, and E. Yahav. Thinking like transformers. In M. Meila and T. Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11080–11090. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/weiss21a.html>. (p. 30)
- K. Wen, Y. Li, B. Liu, and A. Risteski. Transformers are uninterpretable with myopic methods: a case study with bounded dyck grammars. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 38723–38766. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/79ba1b827d3fc58e129d1cbfc8ff69f2-Paper-Conference.pdf. (p. 25)
- N. Wichers, C. Denison, and A. Beirami. Gradient-based language model red teaming, 2024. (p. 18)
- T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush. Transformers: State-of-the-art natural language processing. In Q. Liu and D. Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL <https://aclanthology.org/2020.emnlp-demos.6>. (p. 29)
- B. Wright and L. Sharkey. Addressing feature suppression in saes. *AI ALIGNMENT FORUM*, 2024. URL <https://www.alignmentforum.org/posts/3JuSjTzYmZaSeTxKk/addressing-feature-suppression-in-saes>. (p. 16)
- W. Wu, Y. Wang, G. Xiao, H. Peng, and Y. Fu. Retrieval head mechanistically explains long-context factuality. *Arxiv*, 2024a. URL <https://arxiv.org/abs/2404.15574>. (p. 28)
- Z. Wu, K. D’Oosterlinck, A. Geiger, A. Zur, and C. Potts. Causal proxy models for concept-based model explanations. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org, 2023a. (p. 13)
- Z. Wu, A. Geiger, T. Icard, C. Potts, and N. Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 78205–78226. Curran Associates, Inc., 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f6a8b109d4d4fd64c75e94aaf85d9697-Paper-Conference.pdf. (p. 13)
- Z. Wu, A. Arora, Z. Wang, A. Geiger, D. Jurafsky, C. D. Manning, and C. Potts. Reft: Representation finetuning for language models, 2024b. (p. 13, 15, 30, 31)
- Z. Wu, A. Geiger, A. Arora, J. Huang, Z. Wang, N. D. Goodman, C. D. Manning, and C. Potts. pyvene: A library for understanding and improving pytorch models via interventions, 2024c. (p. 30)
- Z. Wu, A. Geiger, J. Huang, A. Arora, T. Icard, C. Potts, and N. D. Goodman. A reply to makelov et al. (2023)’s “interpretability illusion” arguments, 2024d. (p. 12)
- G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis. Efficient streaming language models with attention sinks. *Arxiv*, 2023. URL <https://arxiv.org/abs/2309.17453>. (p. 21)
- S. M. Xie, A. Raghunathan, P. Liang, and T. Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=RdJVFCHjUML>. (p. 31)
- R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T.-Y. Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*. JMLR.org, 2020. (p. 3)
- S. Yang, S. Huang, W. Zou, J. Zhang, X. Dai, and J. Chen. Local interpretation of transformer based on linear decomposition. In A. Rogers, J. Boyd-Graber, and N. Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10270–10287, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.572. URL <https://aclanthology.org/2023.acl-long.572>. (p. 8)
- Y. Yao, P. Wang, B. Tian, S. Cheng, Z. Li, S. Deng, H. Chen, and N. Zhang. Editing large language models: Problems, methods, and opportunities. In H. Bouamor, J. Pino, and K. Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.632. URL <https://aclanthology.org/2023.emnlp-main.632>. (p. 29)

-
- K. Yin and G. Neubig. Interpreting language models with contrastive explanations. In Y. Goldberg, Z. Kozareva, and Y. Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 184–198, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.14. URL <https://aclanthology.org/2022.emnlp-main.14>. (p. 8)
- Q. Yu, J. Merullo, and E. Pavlick. Characterizing mechanisms for factual recall in language models. In H. Bouamor, J. Pino, and K. Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9924–9959, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.615. URL <https://aclanthology.org/2023.emnlp-main.615>. (p. 28)
- Y. Yu, S. Buchanan, D. Pai, T. Chu, Z. Wu, S. Tong, B. D. Haeffele, and Y. Ma. White-box transformers via sparse rate reduction. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL <https://openreview.net/forum?id=THf18hdVxH>. (p. 31)
- Z. Yu and S. Ananiadou. Locating factual knowledge in large language models: Exploring the residual stream and analyzing subvalues in vocabulary space, 2024. URL <https://arxiv.org/abs/2312.12141>. (p. 25)
- M. Yuksekgonul, V. Chandrasekaran, E. Jones, S. Gunasekar, R. Naik, H. Palangi, E. Kamar, and B. Nushi. Attention satisfies: A constraint-satisfaction lens on factual errors of language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=gfFVATffPd>. (p. 28)
- M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 818–833, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10590-1. (p. 18)
- B. Zhang and R. Sennrich. Root mean square layer normalization. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc. (p. 3)
- F. Zhang and N. Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=Hf17y6u9BC>. (p. 9, 10)
- S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. (p. 23)
- Z. Zhao and B. Shan. Reagent: A model-agnostic feature attribution method for generative language models, 2024. (p. 7)
- Z. Zhong, Z. Liu, M. Tegmark, and J. Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 27223–27250. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/56cbfbf49937a0873d451343ddc8c57d-Paper-Conference.pdf. (p. 26)
- B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations (ICLR)*, 2015. (p. 18)
- H. Zhou, A. Bradley, E. Littwin, N. Razin, O. Saremi, J. M. Susskind, S. Bengio, and P. Nakkiran. What algorithms can transformers learn? a study in length generalization. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=AssIuHnmHX>. (p. 31)
- A. Zou, L. Phan, S. Chen, J. Campbell, P. Guo, R. Ren, A. Pan, X. Yin, M. Mazeika, A.-K. Dombrowski, S. Goel, N. Li, M. J. Byun, Z. Wang, A. Mallen, S. Basart, S. Koyejo, D. Song, M. Fredrikson, J. Z. Kolter, and D. Hendrycks. Representation engineering: A top-down approach to ai transparency. *Arxiv*, 2023. URL <https://arxiv.org/abs/2310.01405>. (p. 14, 15, 26)

A MATHEMATICAL NOTATION

Notation	Definition
n	Sequence length
\mathcal{V}	Vocabulary
$\mathbf{t} = \langle t_1, t_2 \dots, t_n \rangle$	Input sequence of tokens
$\mathbf{x} = \langle \mathbf{x}_1, \mathbf{x}_2 \dots, \mathbf{x}_n \rangle$	Input sequence of token embeddings
d	Model dimension
d_h	Attention head dimension
d_{FFN}	FFN dimension
H	Number of heads
L	Number of layers
$\mathbf{x}_i^l \in \mathbb{R}^d$	Residual stream state at position i , layer l
$\mathbf{x}_i^{\text{mid},l} \in \mathbb{R}^d$	Residual stream state at position i , layer l , after the attention block
$f^c(\mathbf{x}) \in \mathbb{R}^d$	Component c output representation at the last position
$f^l(\mathbf{x}) = \mathbf{x}_n^l \in \mathbb{R}^d$	Residual stream state at the last position, layer l
$\mathbf{A}^{l,h} \in \mathbb{R}^{n \times n}$	Attention matrix at layer l , head h
$\mathbf{W}_Q^{l,h}, \mathbf{W}_K^{l,h}, \mathbf{W}_V^{l,h} \in \mathbb{R}^{d \times d_h}$	Queries, keys and values weight matrices at layer l , head h
$\mathbf{W}_O^{l,h} \in \mathbb{R}^{d_h \times d}$	Output weight matrix at layer l , head h
$\mathbf{W}_{\text{in}}^l \in \mathbb{R}^{d \times d_{\text{FFN}}}, \mathbf{W}_{\text{out}}^l \in \mathbb{R}^{d_{\text{FFN}} \times d}$	FFN input and output weight matrices at layer l
$\mathbf{W}_E \in \mathbb{R}^{d \times \mathcal{V} }$ and $\mathbf{W}_U \in \mathbb{R}^{ \mathcal{V} \times d}$	Embedding and unembedding matrices

Table 1: Notation and definitions of the main variables used in this work.

B LINEARIZATION OF THE LAYER NORM

The LayerNorm operates over an input \mathbf{z} as: $\text{LN}(\mathbf{z}) = \frac{\mathbf{z} - \mu(\mathbf{z})}{\sigma(\mathbf{z})} \odot \gamma + \beta$, where μ and σ compute the mean and standard deviation of \mathbf{z} , and γ and β refer to the element-wise transformation and bias respectively. Holding $\sigma(\mathbf{z})$ as a constant, the LayerNorm can be decomposed into $\mathbf{zL} + \beta$, where \mathbf{L} is a linear transformation:

$$\mathbf{L} := \frac{1}{\sigma(\mathbf{z})} \begin{bmatrix} \gamma_1 & 0 & \dots & 0 \\ 0 & \gamma_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \gamma_n \end{bmatrix} \begin{bmatrix} \frac{n-1}{n} & -\frac{1}{n} & \dots & -\frac{1}{n} \\ -\frac{1}{n} & \frac{n-1}{n} & \dots & -\frac{1}{n} \\ \dots & \dots & \dots & \dots \\ -\frac{1}{n} & -\frac{1}{n} & \dots & \frac{n-1}{n} \end{bmatrix}. \quad (26)$$

C FOLDING THE LAYER NORM

Any Transformer block reads from the residual stream by normalizing before applying a linear layer (with weights \mathbf{W} and bias \mathbf{b}) to the resulting vector:

$$\text{LN}(\mathbf{z})\mathbf{W} + \mathbf{b} \quad (27)$$

Following the decomposition in Equation (26) we can fold the weights of the LayerNorm into those of the subsequent linear layer as follows:

$$\begin{aligned} \text{LN}(\mathbf{z})\mathbf{W} + \mathbf{b} &= (\mathbf{zL} + \beta)\mathbf{W} + \mathbf{b} \\ &= \mathbf{zLW} + \beta\mathbf{W} + \mathbf{b} \\ &= \mathbf{zW}^* + \mathbf{b}^*, \end{aligned} \quad (28)$$

where the new weights and bias are $\mathbf{W}^* = \mathbf{LW}$ and $\mathbf{b}^* = \beta\mathbf{W} + \mathbf{b}$ respectively.

D IMPLEMENTATION DETAILS OF SAEs

- During training, a feature receives a zero gradient signal if it does not activate. When this occurs frequently, it can lead to a dead feature. Bricken et al. (2023) propose **resampling** these features by reinitializing their encoder and decoder weights periodically during

training. An alternative approach to resampling is **ghost gradients** Jermyn & Templeton (2024), which adds an auxiliary loss term that supplies a gradient signal to promote the reactivation of dead features. However, recent results have found this approach suboptimal (Rajamanoharan, 2024; Conerly et al., 2024).

- Setting the β_1 parameter of Adam to 0 has been found to reduce the number of “dead” features in larger autoencoders (Templeton et al., 2024; Rajamanoharan et al., 2024). Yet, Conerly et al. (2024) rely on $\beta_1 = 0.9$.
- Although initially the **norm of the decoder’s rows**²⁴ was recommended to be equal to one (Bricken et al., 2023), recent released SAEs also consider an unconstrained norm setting (Conerly et al., 2024).

²⁴Note that we consider the decoder weight matrix $\mathbf{W}_{\text{dec}} \in \mathbb{R}^{m \times d}$.