How Do LLMs Generate Contrastive Sentiments? A Mechanistic Perspective

Van Bach Nguyen
Marburg University, Germany
vanbach.nguyen@uni-marburg.de

Jörg Schlötterer
Marburg University, Germany
joerg.schloetterer@uni-marburg.de

Christin Seifert
Marburg University, Germany
christin.seifert@uni-marburg.de

Abstract

This paper presents a mechanistic investigation of how large language models (LLMs) generate contrastive sentiments. We define this task as transforming the sentiment of a given text (e.g., from positive to negative) while making minimal changes to its content. We identify two core mechanisms: (1) a preservation mechanism that maintains the sentiment of the input text, primarily mediated by specific attention heads, and (2) a sentiment transformation mechanism, which integrates a representation of the target sentiment label with the original valenced words using a circuit containing both MLP and attention layers. Building on these findings, we propose and validate a novel mechanistic intervention. By modifying key attention heads, we steer the LLM toward more effective contrastive generation, increasing the sentiment flip rate without sacrificing the minimality of changes. Our work not only deepens the understanding of the mechanisms underlying contrastive sentiment generation in LLMs, but also introduces a promising new direction to steer LLM behavior via targeted, mechanistic interventions.[1]

1 Introduction

Large language models (LLMs) achieve impressive performance across a wide range of tasks without task-specific fine-tuning, exhibiting emergent abilities in areas such as translation, classification, and complex reasoning (Brown et al., 2020; Wei et al., 2022). Despite extensive research, the mechanisms underlying these abilities remain poorly understood (Rogers et al., 2020; Rauker et al., 2023; Li et al., 2023). This lack of mechanistic insight limits our ability to predict, control, or enhance LLM performance, particularly on complex tasks.

Inspired by prior work on mechanistic interpretability for specific tasks (Hanna et al., 2023; Feng and Steinhardt, 2024), we investigate how LLMs generate contrastive sentiment for a given text. Contrastive sentiment generation flips the sentiment of a text from positive to negative (or vice versa) while making the fewest possible edits. This task is important in NLP for applications such as model interpretability, adversarial robustness, and data augmentation (Ross et al., 2021; Wu et al., 2021). Although LLMs can generate contrastive outputs (Nguyen et al., 2024b), the underlying mechanisms remain largely unexplored. Understanding these mechanisms not only deepens our insight of LLM behavior, but also enables precise interventions for specific tasks (Bereska and Gavves, 2024).

We investigate the mechanisms underlying contrastive text generation in LLMs through a three-step process (Figure 1). First, we identify the key components that contribute to contrastive outputs. Second, we analyze the interactions among these components, uncovering two primary mechanisms: a *sentiment preservation mechanism* and a *sentiment transformation mechanism*. Third, we perform mechanistic interventions on the identified mechanisms. This final step both validates the findings of the second step and improves performance on the contrastive sentiment generation task.

The two mechanisms uncovered in the second step compete with each other: the *sentiment preservation mechanism* maintains the input text via a group of specific attention heads, while the *sentiment transformation mechanism* alters the sentiment of valenced words, i.e., words that carry positive or negative sentiment[2]. Using a circuit discovery algorithm based on prior work (Dunefsky et al., 2024), we identify the transformation circuit corresponding to the sentiment transformation mechanism. This circuit integrates the target sen-

[1]The source code is available at https://github.com/bach1292/contrastive_generation_MI

[2]We follow the definition from previous work (Tigges et al., 2024; Guerini et al., 2008)

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6619–6635

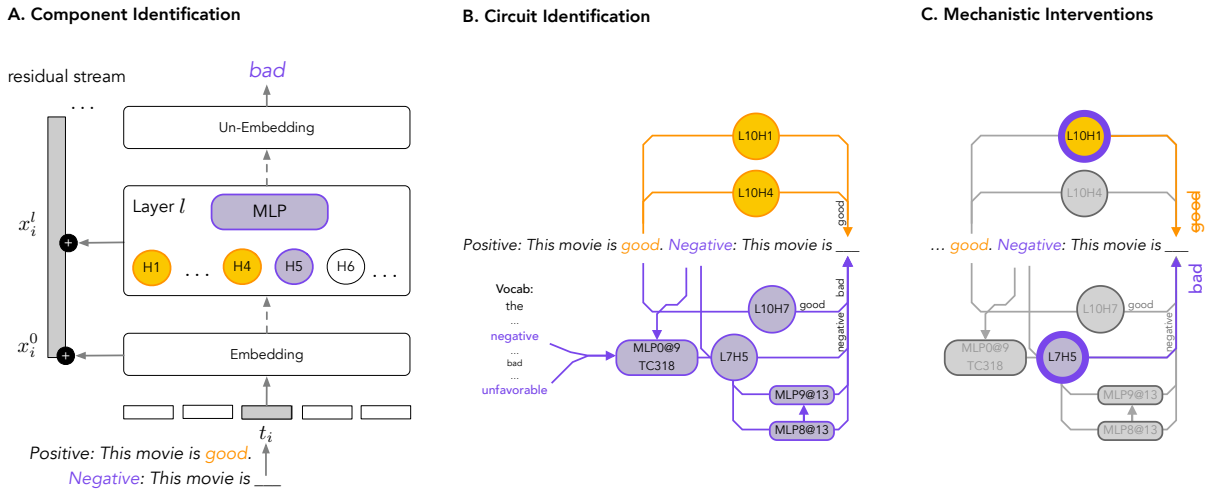


Figure 1: Overview of two mechanisms in LLMs for contrastive text generation: a preservation mechanism driven by a group of specific attention heads (orange) and a sentiment transformation mechanism (purple) involving circuits of MLPs and attention heads. The analysis process consists of three steps: (**A**) identifying key components for the task, (**B**) uncovering the circuit associated with specific sentiments, and (**C**) performing mechanistic interventions to validate and apply findings to real-world tasks. (**B**) and (**C**) show the specific components for the identified circuits in GPT-2. The preservation mechanism (orange) contains attention heads L10H1 and L10H4, which preserve the valenced word. In contrast, the transformation mechanism relies on both, MLPs and attention heads, combining the valenced token *good* and contrastive label token *Negative*. MLP layers can influence one another (e.g., MLP8@13→MLP9@13) and can be decomposed into interpretable transcoder features. For instance, de-embedding transcoder feature MLP0@9TC318 shows that this feature mainly activates on negative tokens. Mechanistic interventions on heads L10H1 or L7H5 suppress the preservation and amplify the transformation mechanism in (**C**) to steer the output.

timent label with the original valenced words and comprises both MLP layers and attention heads. We further find that MLP layers are primarily responsible for encoding the contrastive sentiment label. Building on these insights, we propose a novel mechanistic intervention to guide LLMs in contrastive generation. Our intervention increases the rate of successful contrastive sentiment generation while maintaining a high degree of similarity to the original text. This demonstrates a promising new direction for steering LLM behavior through precise mechanistic interventions. In summary, our contributions are:

1. We conduct a detailed mechanistic analysis of contrastive generation in large language models, identifying two core mechanisms: a *sentiment preservation* and a *sentiment transformation*.

2. We show that *sentiment preservation* is primarily mediated by a small set of attention heads, while *sentiment transformation* arises from a dedicated circuit that integrates valenced words with contrastive labels to produce sentiment-reversed outputs.

3. We propose a novel mechanistic intervention that enables targeted steering of LLMs during contrastive sentiment generation.

2 Related Work

Contrastive Text Generation. Contrastive generation methods aim to minimally alter text while flipping a label. Early approaches relied on token substitutions (Wu et al., 2021; Ross et al., 2021). Recent studies benchmark LLMs on generating sentiment counterfactuals, highlight challenges in content preservation (Nguyen et al., 2024a). We contribute by focusing not only on outcomes but also on the mechanisms inside LLMs that enable such transformations and by using that insight to guide transformations.

Mechanistic Interpretability in LLMs. Mechanistic interpretability decomposes LLMs into interpretable circuits explaining emergent behaviors. Based on the transformer-circuit framework (Elhage et al., 2021), prior work has identified induction and deduction heads (Olsson et al., 2022; Brinkmann et al., 2024) and neuron-level algorithms for arithmetic and logic (Nanda et al., 2023b;

Chughtai et al., 2023). MLPs act as key–value memories (Geva et al., 2021) but suffer from polysemanticity, which sparse autoencoders (Huben et al., 2024) and transcoders (Dunefsky et al., 2024) address by isolating monosemantic features.

Recent methods analyze mechanisms through layer-wise predictions. The logit lens (Nostalgebraist, 2020) projects hidden states through the unembedding layer to track token distributions, enabling analysis of MLPs and attention behaviors (Geva et al., 2023; Dar et al., 2023). Circuit discovery decomposes networks into attention heads and MLPs, identifies sparse subgraphs that perform specific functions, validates them through causal interventions, and evaluates component importance via ablations and logit differences (Tigges et al., 2024; Hanna et al., 2023; Conmy et al., 2023).

Based on these mechanistic interpretability methods, some studies have investigated and explained LLMs from a mechanistic perspective. Hanna et al. (2023) interpret how GPT-2 performs mathematical reasoning, such as computing *greater than*. Ortu et al. (2024) trace how language models handle facts and counterfactuals, showing the interplay between factual knowledge recall and counterfactual statement comprehension. Meanwhile, Tigges et al. (2024) demonstrate that sentiment is linearly represented in LLMs: it is largely captured by a single direction in activation space, so moving along this vector smoothly shifts the model's internal sentiment from negative to positive (or vice versa).

We also adopt existing mechanistic interpretability techniques, specifically logit-lens analysis and transcoders, to identify the layers and circuits responsible for contrastive sentiment generation.

3 Background and Notation

Transformers. Given a text sequence of k tokens, LLMs map each token (along with its position) to a vector representation by applying an embedding matrix $W_E \in \mathbb{R}^{|V| \times d_{\text{model}}}$, where $|V|$ is the vocabulary size and d_{model} is the model dimensionality. This produces the initial *residual stream* $\mathbf{x}_i^0 \in \mathbb{R}^{d_{\text{model}}}$ for each token position $i = 1, \ldots, k$. The residual stream at each position is then updated through a series of L transformer layers, each containing an attention sublayer $\mathbf{Attn}^l$ and an MLP sublayer $\mathbf{MLP}^l$. For layer $l \in \{1, \ldots, L\}$ and token position i, the residual stream is updated as follows:

$$\mathbf{x}_i^l = \mathbf{x}_i^{l-1} + \mathbf{Attn}^l + \mathbf{MLP}^l$$

where $\mathbf{Attn}^l \in \mathbb{R}^{d_{\text{model}}}$ is the output of the attention sublayer and $\mathbf{MLP}^l \in \mathbb{R}^{d_{\text{model}}}$ is the output of the MLP sublayer, both added to $\mathbf{x}_i^{l-1}$.

Contrastive Generation. Given the original text $\boldsymbol{S} := (t_1, \ldots, t_k)$, where t_1 is the label (e.g., "Positive: This movie is good"), the corresponding contrastive text is $\boldsymbol{S'} := (t'_1, t_2, \ldots, t'_k)$, where t'_1 is the contrastive label and t'_k is the contrastive word (e.g., "Negative: This movie is bad"). We concatenate the original text $\boldsymbol{S}$ and the prefix of the contrastive text up to but excluding the final word ($\boldsymbol{S}'_{1:k-1}$) with a period separator, yielding the full prompt $\boldsymbol{S}.\boldsymbol{S}'_{1:k-1}$. The LLM is then prompted to generate the missing final token t'_k.

Next-Token Prediction. Following prior mechanistic interpretability work (Olah et al., 2020; Elhage et al., 2021; Olsson et al., 2022; Nanda et al., 2023a; Ortu et al., 2024), we focus on the next-token prediction task of autoregressive LLMs. For a sequence, we denote the logit for token t_k given the preceding tokens $t_{<k}$ as $T(t_k \mid t_{<k})$. In our setting, the LLM predicts the next token for the prompt $\boldsymbol{S}.\boldsymbol{S}'_{1:k-1}$ (e.g., *Positive: The movie is good. Negative: The movie is* ___). By analyzing the logits (and their changes) for tokens corresponding to the original and contrastive valenced words, we identify the internal components responsible for sentiment transformation.

4 Methodology

In contrastive sentiment generation, the goal is to produce text with the opposite sentiment while suppressing the probability of the original sentiment. For example, given the input *Positive: The movie is good. Negative: The movie is* ___, we expect negative continuations such as *bad* or *terrible*, rather than positive words like *good* or *great*.

Quantifying Preference for Output Tokens. Following prior work (Ortu et al., 2024; Conmy et al., 2023), we quantify the model's preference using the **logit difference** between a desired token t_{desired} and an undesired token $t_{\text{undesired}}$ as $\Delta_T = T_{t_{\text{desired}}} - T_{t_{\text{undesired}}}$ where T denotes the unnormalized output logits. **Desired tokens** align with the target sentiment (e.g., *bad* for negative sentiment), while **undesired tokens** express the opposite (e.g., *good*). A positive Δ_T indicates **success** (the model favors the desired token), and a negative one indicates **failure**.

To uncover the mechanisms underlying contrastive sentiment generation, we first identify which layers and attention heads strongly affect the logit difference Δ_T. We further refine and extend the analysis by identifying circuits, i.e., disjoint sets of components and their interaction, responsible for contrastive sentiment generation. We validate our findings by mechanistic interventions on circuit components. We describe each step in detail below.

4.1 Component Identification

We employ the logit lens (Nostalgebraist, 2020) to track how Δ_T evolves across layers and sublayers, identifying those with the strongest influence. For an intermediate residual stream $\mathbf{x}_i^l$ after a given sublayer at position i and layer l, we compute its scalar projection onto the logit-difference direction:

$$\mathbf{d} = W_U(t_{\text{desired}}) - W_U(t_{\text{undesired}}),$$

where $W_U \in \mathbb{R}^{d_{\text{model}} \times |V|}$ is the unembedding matrix. The projection $\langle \mathbf{x}_i^l, \mathbf{d} \rangle$ quantifies the sublayer's contribution to the final logit difference.

4.2 Circuit Identification

We use transcoders (Dunefsky et al., 2024; Templeton et al., 2024a) to identify sparse subgraphs in the model responsible for sentiment transformation. Transcoders approximate the output of an MLP layer (including the transformation applied by that layer) as a sparse linear combination of interpretable transcoder features. This mitigates the issue of polysemanticity (Bricken et al., 2023; Elhage et al., 2022) in the original MLP, where neurons activate for multiple unrelated concepts. Specifically, for layer l, token index i, and input $\mathbf{x}_i^l \in \mathbb{R}^{d_{\text{model}}}$, the output is:

$$\text{TC}^l(\mathbf{x}_i^l) = \mathbf{W}_{\text{dec}} \mathbf{z}_{\text{TC}}(\mathbf{x}_i^l) + \mathbf{b}_{\text{dec}},$$

$$\mathbf{z}_{\text{TC}}(\mathbf{x}_i^l) = \text{ReLU}(\mathbf{W}_{\text{enc}} \mathbf{x}_i^l + \mathbf{b}_{\text{enc}}),$$

where $\mathbf{W}_{\text{dec}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{features}}}$ and $\mathbf{b}_{\text{dec}} \in \mathbb{R}^{d_{\text{model}}}$ are the decoder matrix and bias, $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{d_{\text{features}} \times d_{\text{model}}}$ and $\mathbf{b}_{\text{enc}} \in \mathbb{R}^{d_{\text{features}}}$ are encoder matrix and bias, and $d_{\text{features}} \gg d_{\text{model}}$. Transcoders are wide ReLU MLPs with one hidden layer, trained with a loss function that balances faithfulness and sparsity:

$$\mathcal{L}_{\text{TC}}(\mathbf{x}) = \|\text{MLP}^l(\mathbf{x}) - \text{TC}^l(\mathbf{x})\|_2^2 + \lambda_1 \|\mathbf{z}_{\text{TC}}(\mathbf{x})\|_1$$

Once we have obtained the MLP transcoders, we then apply a circuit-discovery algorithm (Dunefsky et al., 2024) that starts from a target late-layer feature, attributes upstream contributions from earlier features or attention heads, greedily prunes to the top-k contributors, and recurses backward to build multi-hop circuits down to the embedding layer. Here, we apply *de-embeddings* technique (Dunefsky et al., 2024) to show what kinds of words or concepts a feature tends to represent, giving an input-independent view of its meaning.

4.3 Mechanistic Interventions

After localizing the attention heads responsible for a certain task (e.g., responsible for contrastive sentiment), we intervene on these heads by scaling the value vectors at specific token positions to amplify the effect of important heads (Yu et al., 2023). Formally, let $p \in \{1, 2, \ldots, k\}$ denote a token position whose value vector we intervene. Given a value matrix from an attention sublayer $V \in \mathbb{R}^{k \times d_v}$, where d_v is the value dimension, we construct a modified value matrix V' by scaling the value vector at position p:

$$V'_{i,j} = \begin{cases} \alpha \cdot V_{i,j} & \text{if } i = p \\ V_{i,j} & \text{otherwise} \end{cases}$$

for all $i = 1$ to k and $j = 1$ to d_v. The scalar α adjusts the contribution of the token at position p to the attention output (e.g, $\alpha > 1$ amplifies its influence.) For transformation, p corresponds to the position of the contrastive sentiment label (e.g., *Negative*). For preservation, p corresponds to the position of the original valenced adjective (e.g., *good*). We identify p by tracing high-activation positions associated with the preservation or transformation mechanisms, such as attention patterns in the heads L10H1 and L10H4 for preservation (see Figure 4).

5 Experimental Setup

We study contrastive sentiment generation with two transformation mechanisms under three prompt types. An overview of the experimental conditions is shown in Table 1.

Sentiment Transformation Mechanisms. We study two transformation mechanisms: adjective *Substitution* with antonyms and the addition of *Negation*.

The **substitution** mechanism flips sentiment by replacing a sentiment-bearing adjective with its antonym (e.g., *good* → *bad*). Adjectives are strong

	Minimal Prompt	Contextual Prompt	Instructional Prompt
Template	s_{ori}: w_{ori}. s_{con}:___	s_{ori}: <*context*> w_{ori}. s_{con}: <*context*> ___	<*Instruction*>. s_{ori}: <*context*> w_{ori}. s_{con}: <*context*> ___
Substitution (Example)	Positive: good. Negative: ___	Positive: This movie is good. Negative: This movie is ___	Change the sentiment of the following sentence with minimal changes. [**contextual prompt**]
Negation (Example)	Positive: good. Negative: not ___	Positive: This movie is good. Negative: This movie is not ___	Change the sentiment of the following sentence with minimal changes. [**contextual prompt**]

Table 1: Examples of the three prompt types and two transformation mechanisms. Here, $s_{ori} \in \mathbf{S}$, where $\mathbf{S} = \{$"Positive", "Negative"$\}$ is the original sentiment label, and w_{ori} is a word reflecting s_{ori}. The contrastive sentiment label $s_{con} \in \mathbf{S}$ is the opposite sentiment ($s_{con} \neq s_{ori}$).

carriers of sentiment in natural language. Substituting them with antonyms directly flips the evaluative polarity of the text while keeps the sentence mostly the same. For substitution, the desired output w_{desired} is the contrastive word w_{con}, while the undesired output is the original word w_{ori}.

The **negation** mechanism inverts sentiment by prepending *not* to the adjective (e.g., *good* $\rightarrow$ *not good*). This syntactic modification is a natural and minimal way to reverse polarity, requiring only the addition of one word and no external antonym resources. For negation, the desired output is w_{ori} (the original adjective), while the undesired output is w_{con}, since the addition of *"not"* already inverts the polarity of the original word.

Prompt Types. We use three prompt types to systematically evaluate how different levels of guidance influence the model's ability to perform sentiment transformation.

Minimal prompt: A concise prompt containing only sentiment labels and valenced words. This setting evaluates whether the model can infer the task from minimal input.

Contextual prompt: An extended version of the minimal prompt with simple context added. This tests whether basic contextual information improves the model's ability to perform sentiment transformation.

Instructional prompt: A contextual prompt with an explicit instruction. This setup analyzes the effect of clearly defined task guidance.

Metric. The combination of two mechanisms (substitution and negation) under three different prompt types form six groups in total. Within each group, we compute the **average logit difference** $\overline{\Delta}_T$ across all prompts with different word pairs. These group-wise averages are used to analyze changes in the model's behavior in different setups.

Data. To construct word pairs $(w_{\text{desired}}, w_{\text{undesired}})$, we use a contextual prompt to extract the top-10 adjectives with the highest probabilities, where w_{ori} and w_{con} are *good* and *bad*, respectively, and vice versa (e.g, *Negative/Positive*: This movie is *bad/good*. *Positive/Negative*: This movie is ___.). This yields two adjective sets: positive words $\mathbf{W}_{pos}$ and negative words $\mathbf{W}_{neg}$. For each prompt type, we consider both sentiment transitions (negative$\rightarrow$positive and positive$\rightarrow$negative) and generate 100 word pairs per direction by pairing adjectives from $\mathbf{W}_{pos}$ and $\mathbf{W}_{neg}$, resulting in 200 pairs per prompt type.

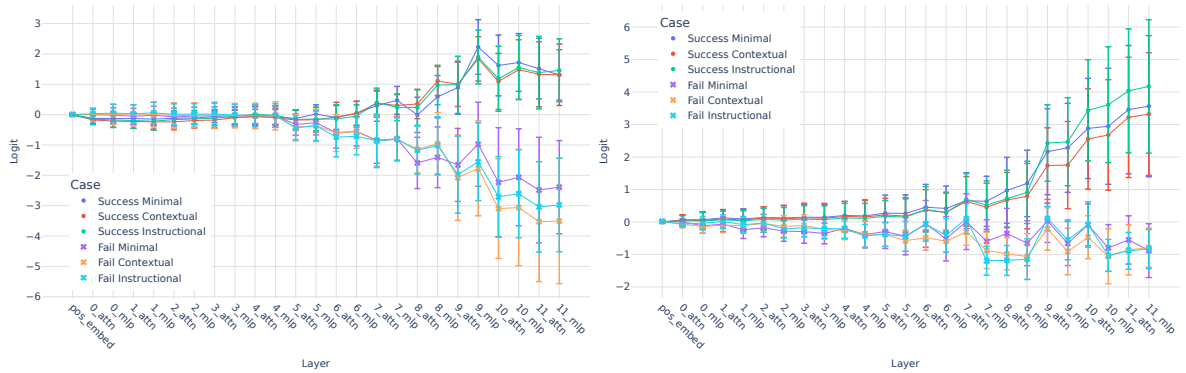
LLMs. Following prior work on mechanistic interpretability (Dunefsky et al., 2024; Ortu et al., 2024), we conduct all experiments using GPT-2 (Radford et al., 2019) and Pythia-410M (Biderman et al., 2023) as representatives of the GPT and Pythia families, respectively. We further analyze heads and layers in Llama-3.2-1B (Dubey et al., 2024) as a representative of the Llama family.

6 Results

We present results first at the layer level, followed by a detailed analysis of individual attention heads, then full circuit discovery, and finally targeted modifications to specific attention heads. Because we observe qualitatively similar patterns across GPT-2, Pythia, and Llama, we focus primarily on GPT-2 in the main text and defer corresponding results for Pythia and Llama to Appendix C and Appendix E, respectively.

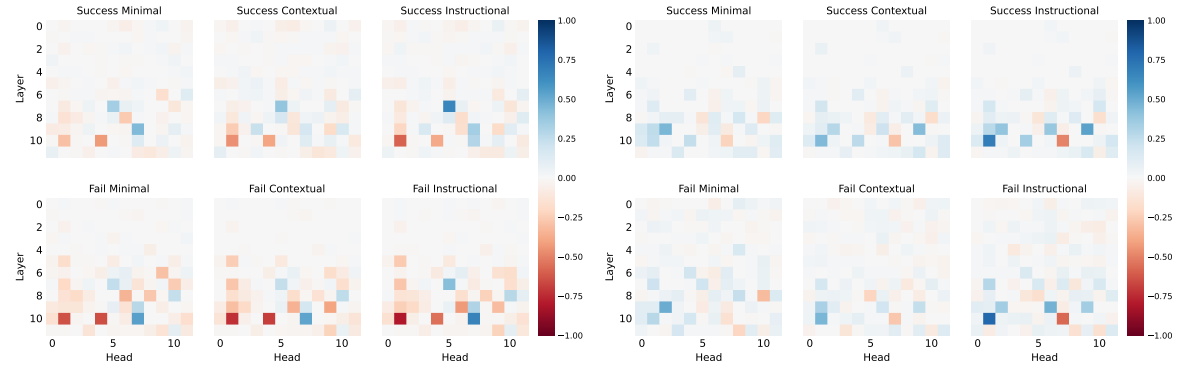
6.1 Layer Analysis

Using Logit Lens (Nostalgebraist, 2020), we measured the logit difference ($\overline{\Delta}_T$) after each sublayer. As shown in Figure 2, $\overline{\Delta}_T$ begins to diverge between success and failure cases from the middle layers onward (starting around layer 6) in both the



(a) Substitution (b) Negation

Figure 2: Accumulative logit difference with standard deviation across layers for the Substitution and Negation.



(a) Substitution (b) Negation

Figure 3: $\overline{\Delta}_T$ over attention heads per layer in Substitution and Negation mechanisms.

substitution and negation settings. This is consistent with prior findings that certain abstract concepts begin to emerge in mid-to-late layers (Jin et al., 2025; Templeton et al., 2024b). In the substitution case, attention sublayers generally contribute negatively to $\overline{\Delta}_T$, whereas MLP sublayers contribute positively (Figure 2a). The pattern reverses in the negation case, where attention sublayers (especially layers 9–10) and MLPs (layers 8–9) drive the largest positive contributions (Figure 2b).

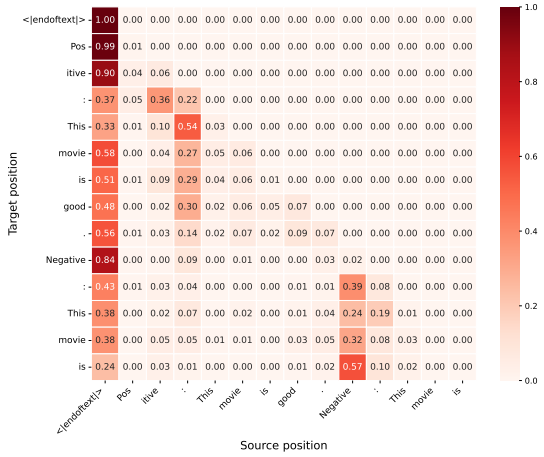
A few attention sublayers deviate from this pattern (e.g., attention sublayer 7 supports substitution success). Across Minimal, Contextual, and Instructional prompt types, patterns stay consistent, though $\overline{\Delta}_T$ is slightly higher for Instructional success cases, indicating a stable core mechanism across different prompt types.

***Takeaway*:** MLP layers typically transform the sentiment of valenced words, while attention sublayers tend to preserve it, though not universally. These patterns remain consistent across different transformation mechanisms (substitution vs. negation) and prompt types.

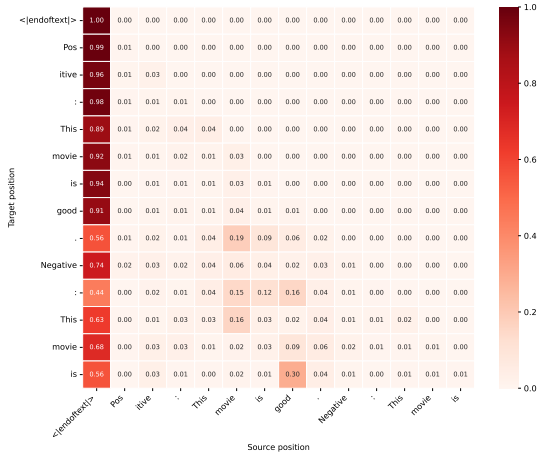
6.2 Attention Heads Analysis

Figure 3 shows the average $\overline{\Delta}_T$ across all attention heads per layer, highlighting the importance of each head in each layer. Certain heads significantly influence $\overline{\Delta}_T$, contributing either positively or negatively. In the *substitution* case, Head 5 in Layer 7 (L7H5), L9H7, and L10H7 consistently show positive contributions to $\overline{\Delta}_T$ in both success and failure cases (see Figure 3a). Interestingly, L7H5 contributes strongly to positive $\overline{\Delta}_T$ in success cases, while L10H7 does so in failure cases. In contrast, L10H1 and L10H4 contribute strongly to negative $\overline{\Delta}_T$. In the *negation* case, the effects of these heads are mostly inversed.

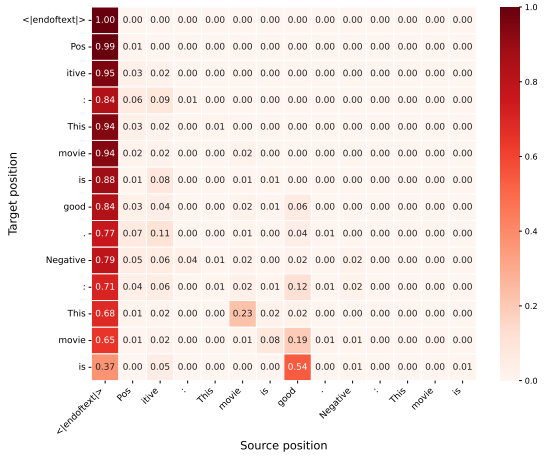
Analysis of Important Heads. Our analysis of the attention patterns in these heads shows that L7H5 moves information from the contrastive sentiment label s_{con} to the last token (Figure 4a), while L9H7 transfers information from the original valenced word w_{ori} to the last token (Figure 4b). Both heads contribute positively to $\overline{\Delta}_T$. In contrast, L10H1 also moves information from w_{ori} (Fig-



(a) L7H5



(b) L9H7



(c) L10H1

Figure 4: Attention patterns of heads L7H5, L9H7 and L10H1, illustrating their respective information-transfer roles in GPT-2.

ure 4c), but contributes negatively to $\overline{\Delta}_T$. L10H7 behaves similarly to L9H7 and L10H4 similarly to L10H1 (see Figure 8 in the appendix for their attention heatmaps). This pattern suggests that the information from the valenced word w_{ori} (e.g., *good*) is critical, as it helps identify the next token but can also lead the model to preserve the original sentiment s_{ori} instead of transforming it. Conversely, Figure 3b shows that heads contributing positively in the *substitution* case tend to contribute negatively in the *negation* case, and vice versa. For example, in the negation case, L10H1 contributes positively to $\overline{\Delta}_T$, indicating that it preserves w_{ori} directly.

***Takeaway*:** Attention heads transfer information from the contrastive sentiment label and the valenced word. Some heads contribute to transformation, some to preservation.

6.3 Circuit Analysis

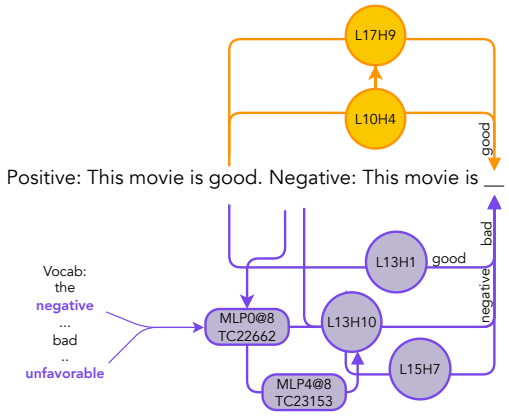
Using the transcoder-based circuit discovery algorithm (Dunefsky et al., 2024), we identify relevant circuits with a **contextual prompt**[3]. Our analysis confirms that MLP layer 9 (MLP9) and attention head L7H5 significantly drive positive $\overline{\Delta}_T$. The transcoder shows that L7H5 at token 9 (L7H5@9), which processes contrastive sentiment s_{con}, strongly influences MLP9@13 and MLP8@13. This aligns with our earlier finding that MLP8 and MLP9 substantially contribute to positive $\overline{\Delta}_T$. Additionally, we identify a circuit path from MLP0@9 to MLP9@13 via L7H5@9. The circuit is depicted in Figure 1B.

MLPs. To interpret MLPs, we used the pretrained transcoder for GPT-2 (Dunefsky et al., 2024), which emulates MLP sublayers and replaces uninterpretable MLP layers with interpretable features. De-embedding of transcoder features shows that MLP0@9 and MLP9@13 contain transcoder features (TC) associated with s_{con}, including *positive* (TC9570) and *negative* (TC318) sentiment (Figure 1B). This confirms a sentiment circuit that involves s_{con} exists. L7H5 plays a critical role in propagating s_{con} information from layer 0 throughout GPT-2.

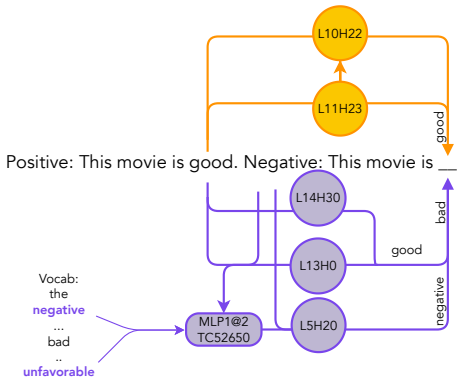
Attention Heads Contribution. Furthermore, we observe that L10H7, which transfers information from the valenced word, also contributes to this circuit. Thus, the LLM integrates information from w_{ori} and s_{con} to generate w_{con} (see Figure 1B).

Circuit Generalization. The transformation circuits in Pythia (Figure 5a) and Llama-3.2-1B (Figure 5b) show slight differences from GPT-2. While

[3]Token order shown in Figure 4



(a) Pythia



(b) Llama

Figure 5: The circuit responsible for sentiment transformation (purple) and the main components of the preservation mechanism (orange) in (a) Pythia and (b) Llama.

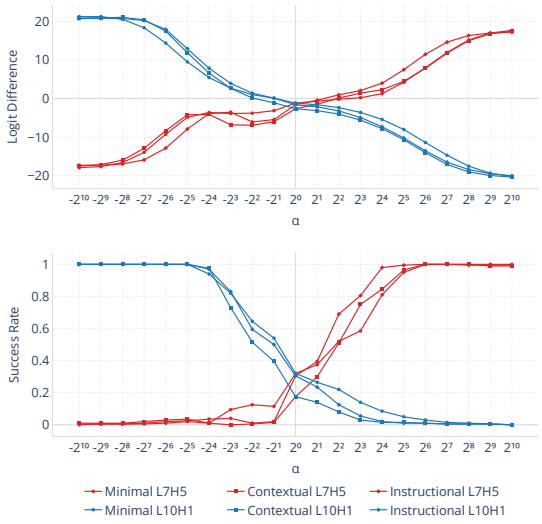


Figure 6: The logit difference (top figure) and success rate (bottom figure) for varying α values applied to the positive head L7H5 (red) and negative head L10H1 (blue) across different prompt types. Applying positive α to the positive head or negative α to the negative head increases both logit difference and success rate, while the converse decreases them, demonstrating the method's effectiveness and confirming the significance of these heads. The baseline is $\alpha = 2^0$, which indicates no modification.

GPT-2 relies heavily on MLPs for transformation, attention seems to play a more important role in the transformation mechanisms of the Pythia and Llama models. For the preservation circuit, we observe similar patterns across all three models.

***Takeaway*:** The transformation mechanism requires both MLP and attention sublayers: MLPs handle the contrastive label, while attention heads transfer information from the valenced word, and their combination forms a new contrastive word. Meanwhile, a subset of attention heads preserves the original sentiment, competing with the transformation mechanism.

6.4 Attention Modification

We aim to improve the LLM's performance on contrastive generation by steering it toward the expected (contrastive) token or by increasing the logit difference between expected and unexpected tokens. To achieve this, after identifying the key attention heads for the task, we apply the attention modification (Section 4.3) to adjust the attention value of a specific head at a specific position, thereby controlling the output as desired.

Intervention Mechanisms. We consider two intervention strategies: (1) amplifying heads that contribute positively to $\overline{\Delta}_T$, and (2) negating heads that contribute negatively to $\overline{\Delta}_T$. Unlike prior work (Ortu et al., 2024; Yu et al., 2023) that only uses positive α to reduce or amplify effects, we propose using a negative α for sentiment analysis tasks. This turns heads causing negative $\overline{\Delta}_T$ into heads that produce positive $\overline{\Delta}_T$. With $\alpha < -1$, we can further amplify this effect. Intuitively, the values with $\alpha < -1$ first inverse the sentiment and then amplify it. This approach is useful for tasks that require a switch between two opposite labels.

Setup. We apply both positive ($\alpha > 0$) and negative ($\alpha < 0$) interventions to the setup in Section 5. We modify L7H5 at the token position of s_{con} and L10H1, as these heads contribute most to positive and negative logit differences (see Figure 3a). We test $\alpha = 2^k, k = 0, 1, \ldots, 10$ and report the aver-

age logit difference $\overline{\Delta}_T$ and the success rate which is measured by the number of pairs having $\Delta_T > 0$ over all pairs.

Results. The results in Figure 6 show that both amplifying a positive head (e.g L7H5, red lines) or negating a negative head (e.g L10H1, blue lines) increases the logit difference and the success rate for all prompt types, demonstrating the effectiveness of these modifications. We also observe that negating positive heads produces negative effects, and $\overline{\Delta}_T$ converges when α is sufficiently large.

Amplifying or negating individual heads is effective, but it remains unclear whether their combination shows stronger effects. To investigate this, we modified two heads simultaneously: L7H5 at the s_{con} position and L10H1 at the w_{ori} position. However, their effects do not combine additively, as outputs from earlier layers propagate to later ones. As a result, the combined modification achieves the same performance as amplifying L7H5 alone. Detailed results are reported in Appendix B.

***Takeaway*:** Amplifying positive heads or negating negative heads can steer LLMs toward the desired sentiment. However, their effects do not combine additively.

7 Validation on Downstream Task

Modern LLMs like GPT-4, Llama-3.3 can generate contrastive sentiment well (Nguyen et al., 2024b), but GPT-2 struggles to produce contrastive examples that flip the label. To test whether attention modification can improve GPT-2's contrastive generation, we set up a realistic contrastive generation task. In this task, the LLM must generate new text that flips the label of a classifier with minimal changes to the original text.

Setup. We use an instructional prompt with sentences from the SST-2 dataset (Socher et al., 2013), but remove all context after contrastive sentiment s_{con} (e.g. *Positive: Such a good movie. Negative:*), requiring the model to generate the entire contrastive sentence. Because sentence lengths and the position of the original valenced word w_{ori} vary, we cannot determine the position of w_{ori}, making it impossible to negate the effect of attention heads focused on it. However, as the label is fixed, we know the position of s_{con} in the prompt in advance, and can amplify the head attending to this token. Similar to previous experiments, we amplify the **L7H5** head using different α values,

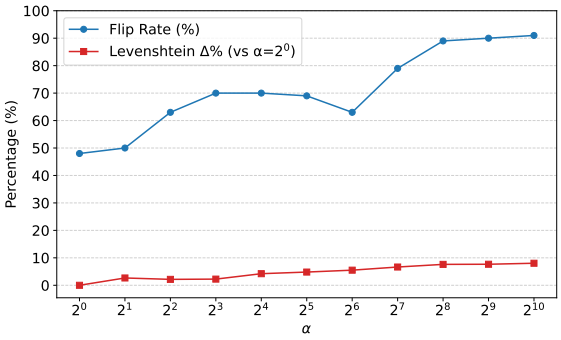


Figure 7: The flip rate, and token-level Levenshtein distance percentage changes compared to the baseline (no intervention, $\alpha = 2^0$) when amplifying L7H5 with varying α values.

$\alpha = 2^k, k = 0, 1, \ldots, 10$. We then measure the flip rate and the token-level edit distance between the original input and the counterfactual output.

Results. Figure 7 shows the effectiveness of our approach: increasing α improves the flip rate (from $< 50\%$ to $> 90\%$) with minimal impact on the token distance ($< 10\%$) compared to the baseline case when $\alpha = 2^0$ at which GPT-2 performs poorly. We observe similar results on Pythia when amplifying L13H10 (see Appendix D).

***Takeaway*:** Mechanistic intervention through attention modification has the potential to improve LLMs' performance on real-world datasets.

8 Conclusion

In this work, we present a mechanistic analysis of contrastive sentiment generation in LLMs, identifying two competing processes: (1) a preservation mechanism via attention heads, and (2) a sentiment transformation circuit integrating target labels with valenced words to invert polarity. Using logit lens, transcoder-based circuit tracing, and targeted interventions, we show that specific attention heads propagate contrastive labels, while MLPs encode sentiment shifts.

Our work not only deepens understanding of how LLMs perform contrastive sentiment generation but also highlights a promising direction for steering LLM behavior via targeted mechanistic interventions. Future work could apply our three-step analysis to larger models, other tasks, and multilingual settings, explore interactions with emergent abilities like reasoning, or develop automated tools for discovering and intervening in similar circuits across diverse tasks.

Limitations

The experiment setup relies on very targeted and simple prompts, even though they cover diverse contexts. The intervention indicates a promising direction for improvement, but may also introduce unintended side effects to the LLMs.

Acknowledgments

We gratefully acknowledge support from the hessian.AI Service Center (funded by the **Federal Ministry of Research, Technology and Space, BMFTR,** grant no. 16IS22091) and the hessian.AI Innovation Lab (funded by the Hessian Ministry for Digital Strategy and Innovation, grant no. S-DIW04/0013/003).

References

Leonard Bereska and Stratis Gavves. 2024. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*. Survey Certification, Expert Certification.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, and 1 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2.

Jannik Brinkmann, Abhay Sheshadri, Victor Levoso, Paul Swoboda, and Christian Bartelt. 2024. A mechanistic analysis of a transformer trained on a symbolic multi-step reasoning task. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4082–4102, Bangkok, Thailand. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. 2024. Transcoders find interpretable llm feature circuits. *Advances in Neural Information Processing Systems*, 37:24375–24410.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Preprint*, arXiv:2209.10652.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Jiahai Feng and Jacob Steinhardt. 2024. How do language models bind entities in context? In *The Twelfth International Conference on Learning Representations*.

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

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

Marco Guerini, Carlo Strapparava, and Oliviero Stock. 2008. Valentino: A tool for valence shifting of natural language texts. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In *Advances in Neural Information Processing Systems*, volume 36, pages 76033–76060. Curran Associates, Inc.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, Fan Yang, Mengnan Du, and Yongfeng Zhang. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 558–573, Abu Dhabi, UAE. Association for Computational Linguistics.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023a. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations*.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023b. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.

Van Bach Nguyen, Christin Seifert, and Jörg Schlötterer. 2024a. CEval: A benchmark for evaluating counterfactual text generation. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 55–69, Tokyo, Japan. Association for Computational Linguistics.

Van Bach Nguyen, Paul Youssef, Christin Seifert, and Jörg Schlötterer. 2024b. LLMs for generating and evaluating counterfactuals: A comprehensive study. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14809–14824, Miami, Florida, USA. Association for Computational Linguistics.

Nostalgebraist. 2020. Interpreting gpt: The logit lens. Blog post.

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

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *arXiv preprint*.

Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. 2024. Competition of Mechanisms: Tracing How Language Models Handle Facts and Counterfactuals. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8420–8436, Bangkok, Thailand. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Tilman Rauker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. 2023. Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks . In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 464–483, Los Alamitos, CA, USA. IEEE Computer Society.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Adly Templeton, Joshua Batson, Adam Jermyn, and Chris Olah. 2024a. Predicting future activations, january 2024. *URL https://transformer-circuits. pub/2024/jan-update/index. html# predict-future*.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy

Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, and 3 others. 2024b. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Curt Tigges, Oskar J. Hollinsworth, Atticus Geiger, and Neel Nanda. 2024. Language models linearly represent sentiment. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 58–87, Miami, Florida, US. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.

Qinan Yu, Jack Merullo, and Ellie Pavlick. 2023. Characterizing mechanisms for factual recall in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9924–9959, Singapore. Association for Computational Linguistics.

A Attention Patterns

We analyze attention patterns in GPT-2 to identify the source and destination of information transfer. Figure 8a shows that the top positive head, L7H5, moves information from the contrastive sentiment label to the final token, whereas L9H7 (Figure 8b) and L10H7 (Figure 8c move information from valenced words (e.g., *good*) to the final token. Both these heads contribute positively to $\overline{\Delta}_T$. While L10H1 (fig. 8d) and L10H4 (fig. 8e) also move information from valenced words to the final token, they instead contribute negatively to $\overline{\Delta}_T$. These patterns support our interpretation that L7H5, L9H7 and L10H7 operate as part of the **transformation mechanism**, while L10H1 and L10H4 operate as part of the **preservation mechanism**.

B GPT-2: Two heads modification

To assess the impact of combining heads, we examine the average $\overline{\Delta}_T$ cross-attention per layer for the Substitution mechanism when amplifying L7H5 and negating L10H1. Figure 9 shows that the effect of amplifying L7H5 propagates to later layers, including Layer 10, thereby decreasing the influence of the negating head.

To directly evaluate this combination, we also compute the logit difference and flip rate, following Section 6.4. Specifically, Figure 10 compares amplifying L7H5 by α (red) with the combination of negating L10H1 by α and amplifying L7H5 by $-\alpha$ (blue).

C Pythia: Results

Following the experiments with GPT-2 (See Section 6), we analyze Pythia-410M (Biderman et al., 2023), which has 24 layers and 16 attention heads.

C.1 Layer Analysis

Substitution mechanism in the Pythia model is similar to GPT-2 (cf. Figure 13a). Starting from layer 16, a clear pattern emerges: attention layers cause a negative logit difference $\overline{\Delta T}$, leading to failure cases, whereas MLP layers cause a positive $\overline{\Delta T}$, leading to success cases. Before this point, between layers 13–15, we see behavior similar to layer 7 in GPT-2, where attention layers contribute to an increase in $\overline{\Delta T}$ while MLP layers have only minimal impact.

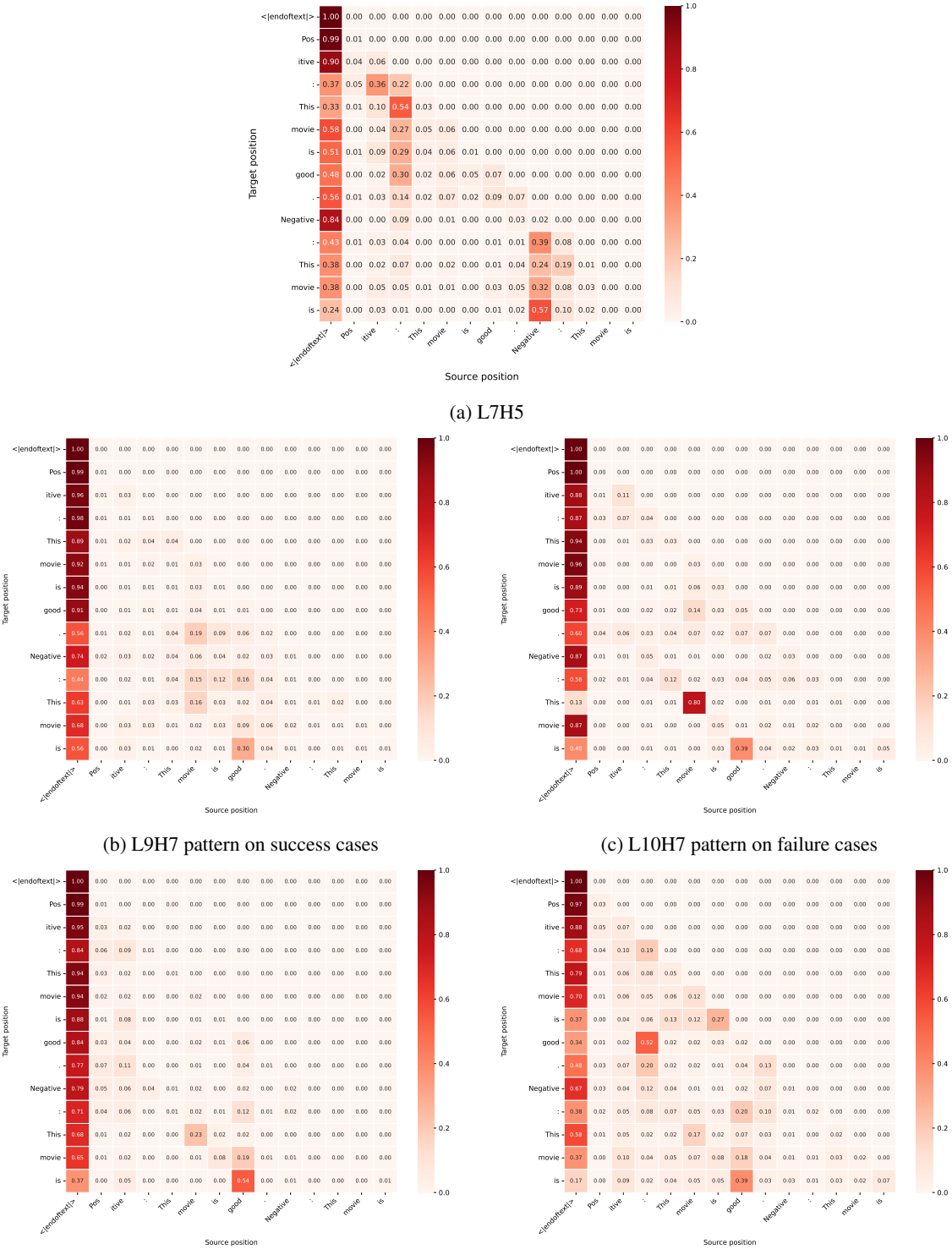
For the negation mechanism, we again observe an opposition to the substitution mechanism, consistent with GPT-2. As shown in Figure 13b, from layer 15 onward, MLP layers contribute to a negative $\overline{\Delta T}$, while attention layers contribute positively. Note that, when using instruction sentences, we do not observe any failure cases.

C.2 Attention Head Analysis

Figure 14a shows a pattern similar to GPT-2. Attention heads in the middle layers (e.g. L13H1, L13H10) contribute positively to $\overline{\Delta T}$ (as seen in layer 7 of GPT-2), whereas attention heads in the later layers contribute negatively (e.g. L17H9). The opposite holds in the negation mechanism (Figure 14b): some heads that show strong negative contributions in the substitution mechanism instead show strong positive contributions under negation.

C.3 Circuit Analysis

Similar to the GPT-2 analysis, we also use the transcoder algorithm (Dunefsky et al., 2024) to



(a) L7H5

(b) L9H7 pattern on success cases

(c) L10H7 pattern on failure cases

(d) L10H1 pattern on success cases

(e) L10H4 pattern on success cases

Figure 8: Attention patterns across important heads, illustrating information-moving roles in GPT-2.

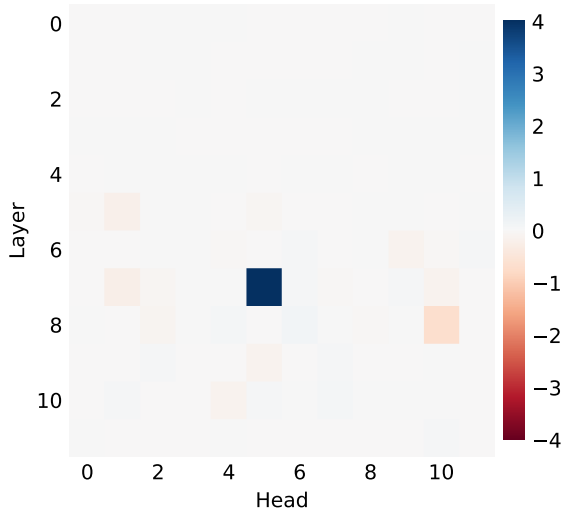


Figure 9: *GPT-2*. Average $\overline{\Delta}_T$ across attention heads per layer for the Substitution mechanism when combining amplifying head L7H5 with negating head L10H1. The effect of amplifying L7H5 propagates to later layers, including Layer 10, thereby reducing the impact of the negating head.

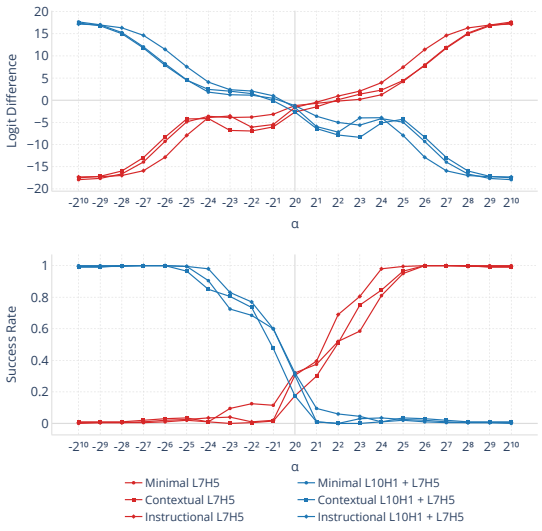


Figure 10: *GPT-2*. Logit difference (top) and success rate (bottom) for varying α values applied either to the positive head L7H5 alone (red) or to the combination of the negative head L10H1 and positive head L7H5 (blue) across different prompt types. For the combination, α is applied to L10H1 and $-\alpha$ to L7H5 (except for the baseline 2^0). The combination does not provide any advantage over amplifying L7H5 alone.

identify the circuits responsible for the transformation mechanism. Figure 5a shows that attention heads such as L13H10 and L15H7 play important roles in transferring information from the

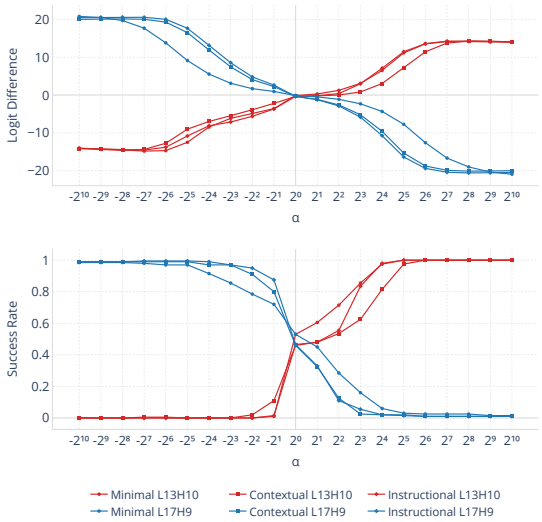


Figure 11: *Pythia*. The logit difference (top figure) and success rate (bottom figure) for varying α values applied to the positive head L13H10 (red) and negative head L17H9 (blue) across different prompt types. Applying positive α to the positive head or negative α to the negative head increases both logit difference and success rate, while the converse decreases them, demonstrating the method's effectiveness and confirming the significance of these heads. The baseline is $\alpha = 2^0$, which indicates no modification.

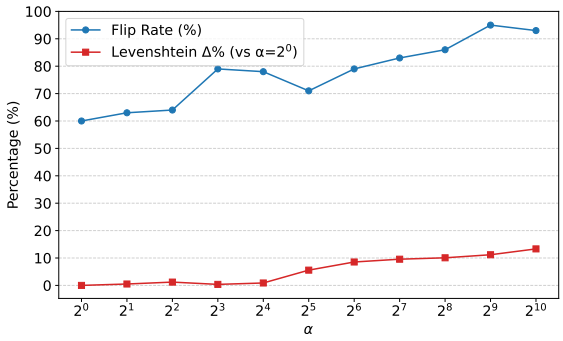
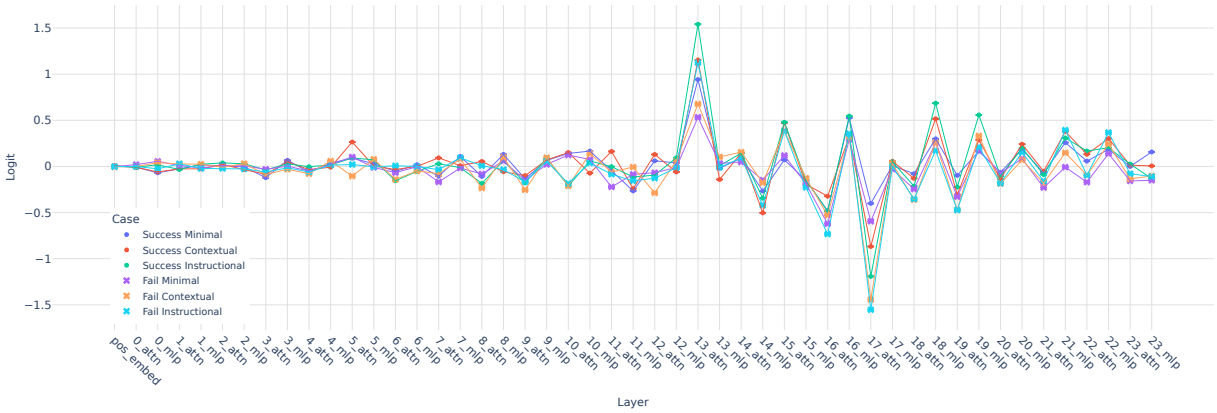
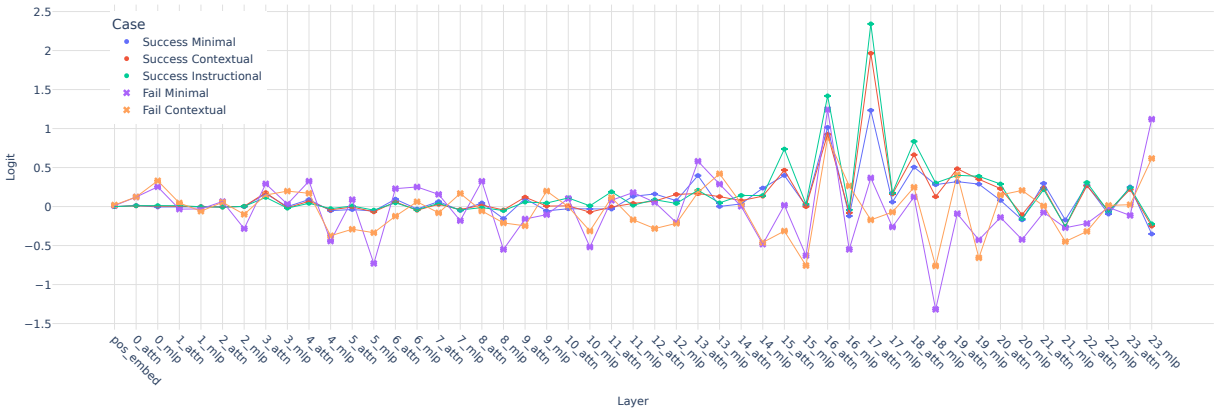


Figure 12: The flip rate, and Levenshtein distance percentage changes compared to the baseline (no intervention, $\alpha = 2^0$) when amplifying L13H10 with varying α values for Pythia.

contrastive label to the final tokens to achieve sentiment transformation. As in GPT-2, information from valenced words is also required, and head L13H1 is responsible for capturing it. These results indicate that our findings generalize beyond GPT-2 to Pythia as well.

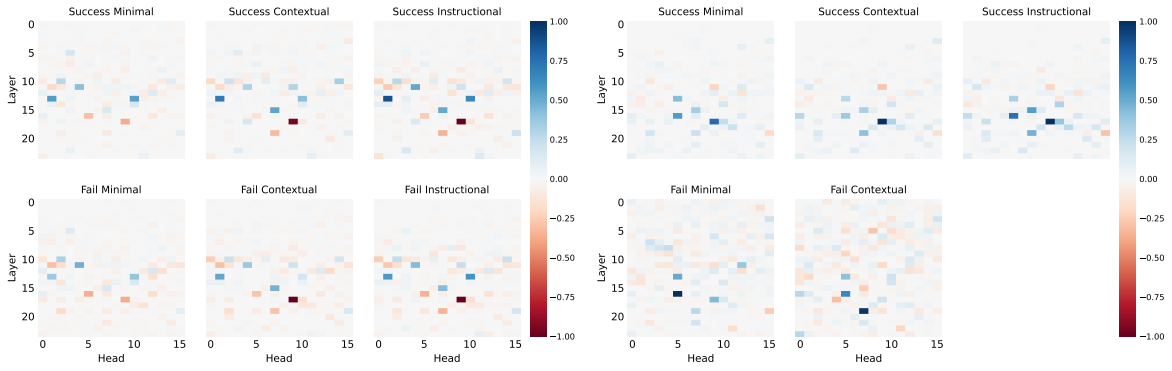


(a) Substitution



(b) Negation

Figure 13: Logit difference per layer in Substitution and Negation mechanisms in Pythia.



(a) Substitution (b) Negation

Figure 14: $\overline{\Delta}_T$ over attention heads per layer in Substitution and Negation mechanisms in Pythia.

C.4 Attention Modification

For Pythia, similar to GPT-2, we identify head L13H10 as part of the transformation circuit, responsible for moving label information and combining it with the MLP layer to transform the valenced word. In contrast, heads such as L17H9 serve a preservation role by maintaining the original sentiment. Following Section 6.4, we apply amplifying and negating techniques to L13H10 and L17H9, respectively. Figure 11 presents the impact of these attention modification methods on logit difference and success rate for the Pythia model.

D Pythia: Validation on Downstream Task

Similar to GPT-2 in Section 7, we validate our amplification techniques on Pythia. In this model, we amplify L13H10 using different α values, where

$\alpha = 2^k$ for $k = 0, 1, \ldots, 10$. The results in Figure 12 show a similar pattern to GPT-2: increasing α improves the flip rate (from 60% to $> 90\%$) with minimal impact on token distance ($< 20\%$), compared to the baseline case when $\alpha = 2^0$.

E Llama: Results

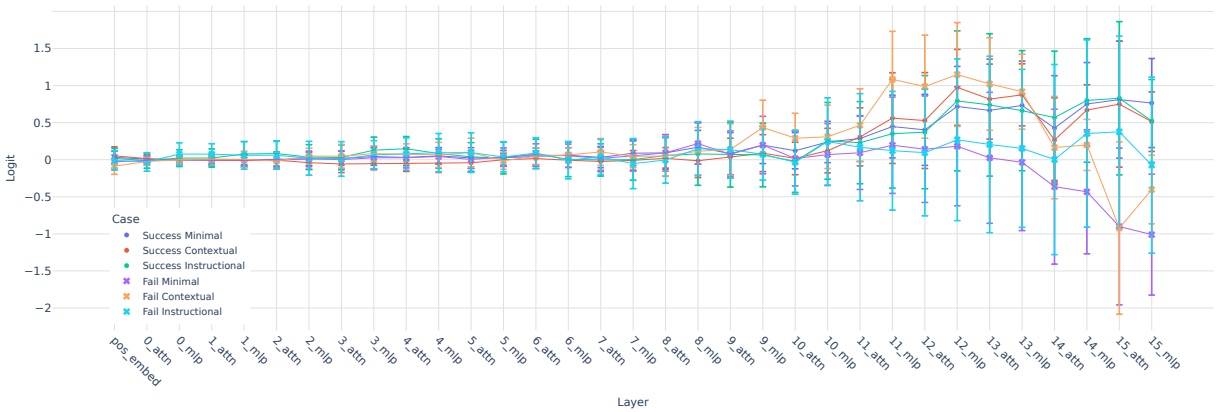
Following the experiments with GPT-2 (See Section 6), we analyze Llama3.2-1B (Dubey et al., 2024), which has 24 layers and 16 attention heads.

As in Pythia and GPT-2, we observe the same pattern in LLaMA-3.2-1B: MLP layers primarily drive the transformation mechanism, while attention heads are mainly responsible for the preservation mechanism in the *substitution* case (see Figure 15). However, the contribution of MLP layers in the LLaMA circuit is weaker than in GPT-2 and Pythia (see Figure 5b).

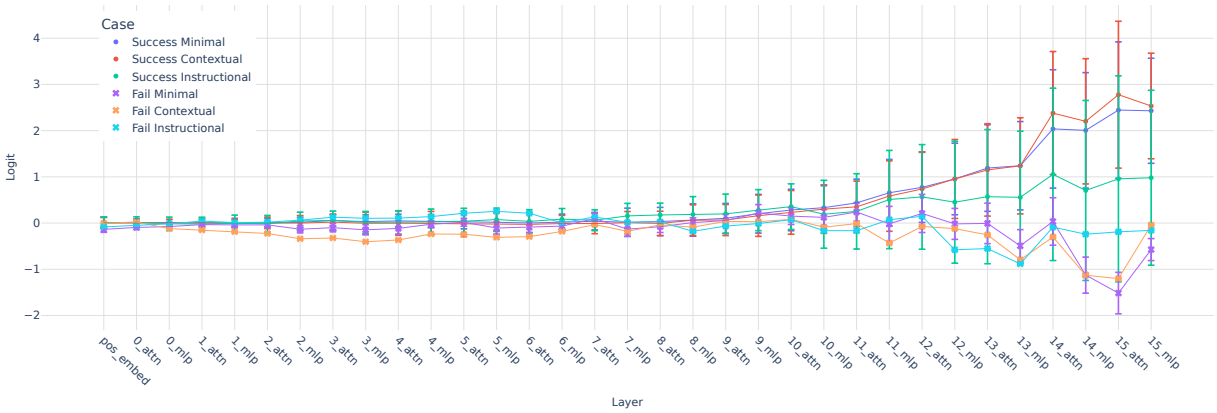
F Computational Resources and Software Use

In this paper, we use the `transformerlens` [4] package for the analysis. We perform the analysis on an RTX 3090 GPU and train the transcoder on an A100 80GB GPU, which takes approximately 8 hours.

[4] https://transformerlensorg.github.io/TransformerLens/

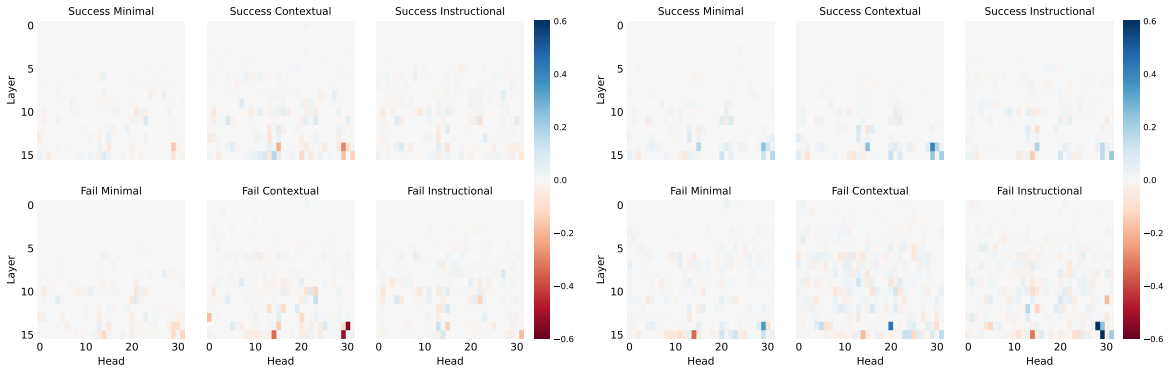


(a) Substitution



(b) Negation

Figure 15: Logit difference per layer in Substitution and Negation mechanisms in Llama.



(a) Substitution

(b) Negation

Figure 16: $\overline{\Delta}_T$ over attention heads per layer in Substitution and Negation mechanisms in Llama.