From Indirect Object Identification to Syllogisms: Exploring Binary Mechanisms in Transformer Circuits

Anonymous ACL submission

Abstract

Transformer-based language models (LMs) can perform a wide range of tasks, and mechanistic interpretability (MI) aims to reverse engineer the components responsible for task completion to understand their behavior. Previous MI research has focused on linguistic tasks like Indirect Object Identification (IOI). In this paper, we investigate the ability of GPT-2 small to handle binary truth values by analyzing its behavior with syllogistic prompts, such as "*Statement A is true. Statement B matches statement A. Statement B is*", which requires more complex logical reasoning compared to IOI. Through our analysis of several syllogism tasks of varying difficulty, we identify multiple circuits that mechanistically explain GPT-2's logical-reasoning capabilities and uncover binary mechanisms that facilitate task completion, including the ability to produce a negated token that does not appear in the input prompt through negative heads. Our evaluation using a faithfulness metric shows that a circuit comprising five attention heads achieves over 90% of the original model's performance. By relating our findings to IOI analysis, we provide new insights into the roles of certain attention heads and MLPs in LMs. We believe these insights contribute to a broader understanding of model reasoning and benefit future research in mechanistic interpretability.

1 Introduction

Despite the success of Large Language Models (LLMs) and their amazing capabilities, these models remain largely opaque and function as black boxes. Mechanistic interpretability has emerged as a field dedicated to mitigate this conceptual gap. By analyzing how LMs solve specific tasks (Wang et al., 2022; Hanna et al., 2023; Merullo et al., 2024), studying emergent behaviors (Arditi et al., 2024), and identifying patterns within their architectures (Gurnee et al., 2024), researchers aim to unravel the inner workings of LMs. Even though great progress has been made, significant gaps remain in understanding LMs even on basic tasks.

GPT-2 is a family of representative LLMs that has been frequently studied in mechanistic interpretability literature. An exemplary case is analyzing its ability to do Indirect Object Identification task (Wang et al., 2022), which reverse engineers how GPT-2 correctly predicts the final token in sentences like "*When Mary and John went to the shops, John gave a bottle of milk to*". Such mechanistic analysis begins with the output and traces back to identify the architectural components relevant to the task, termed as *circuit*. GPT-2 small has been shown to be competent in such linguistic tasks, however its ability and mechanism to perform logic reasoning remains uncertain. Specifically, it lacks the capability to coherently answer true-false questions, such as "*True or False? Dogs have four legs.*". To investigate how GPT-2 represents and processes truth values, we utilize syllogism tasks — a classic form of logical reasoning involving premises and a conclusion. By applying similar mechanistic analysis to syllogistic prompts, we aim to discover the circuits that are relevant to the task and interpret the internal mechanisms GPT-2 uses when handling truth values.

This paper builds on previous interpretability research by focusing on how GPT-2 handles syllogistic prompts. We use true-false syllogism tasks where truth values are assigned to premise statements and the model is prompted to predict the truth value of the conclusion. We define three prompt formats to probe binary reasoning. The *Simple Syllogism* (SS) presents direct entailment, e.g., "*Statement A is true. Statement B matches statement A. Statement B is*". The *Opposite Syllogism* (OS) inverts this logic, requiring negation, e.g., "*Statement A and Statement B are opposite. Statement A is true. Statement B is*". The *Complex Syllogism* (CS) adds one or more distractor premises irrelevant to the inference, e.g., "*Statement A is true. Statement B matches statement A. Statement C is false. Statement B is*", where the distractor is "*Statement C is false*".

Our approach includes two mechanistic interpretability techniques: Path Patching and Logit Lens. Path patching (Wang et al., 2022) determines the importance of a computational component in

solving a task by replacing part of the model's forward pass with activations from a different distribution. Logit Lens (Nostalgebraist, 2020) applies the model's unembedding matrix at different stages of the residual stream, exposing logits and offering insights into the function of specific components during the model's processing. Using these techniques, we apply a mechanistic lens to uncover how LMs perform complex reasoning tasks and identify the key components that drive their decisions. Specifically, we examine the internal mechanisms responsible for **negation** and **reinforcement** of truth values. Evaluation with a circuit faithfulness metric shows that a circuit of three attention heads can recover 90% of the original model's performance on SS prompts. For OS prompts, a circuit of five attention heads and four MLPs nearly recovers the performance of the full GPT-2 model, achieving roughly 85% faithfulness. The structure of the OS circuit is shown in Figure 1.

Throughout our investigation into how GPT-2 processes syllogisms, we uncover several insights into its internal mechanisms and reasoning capabilities. Our contributions include:

1. **Discover Syllogism-Specific Circuits**: We discover circuits that represent the internal mechanisms through which GPT-2 solves syllogisms of varying complexity.

2. **Identify A Negation Mechanism**: We identify a novel mechanism for outputting the **negation** of a truth value. Attention heads suppress the truth logit and MLPs modulate the **negation** of the truth logit in the output distribution.

3. **Explain Importance of Negative Components**: Through analysis of a pair of semantically opposite tasks (SS and OS), we demonstrate that components critical for one task often have corresponding negative counterparts that play a causally important role in the opposite task. This provides new insights into how language models process and represent binary pairs of tokens.

2 Preliminary

Transformers Circuits

We provide a brief overview of GPT-2 following the notation from Elhage et al. (2021). GPT-2 is a decoder-only transformer with 12 layers; each layer contains 12 attention heads and one MLP. Input tokens t are embedded into the initial residual stream state x_0. The residual stream, a core intermediate representation, is updated additively as it passes through each layer's components

At layer i, the residual stream x_{i-1} is processed by the layer's components and updated as follows:

$$\begin{aligned} x_i = x_{i-1} \\ &+ \text{AttentionHeads}(x_{i-1}) \\ &+ \text{MLP}\big(x_{i-1} + \text{AttentionHeads}(x_{i-1})\big). \end{aligned}$$

Here, the attention heads process x_{i-1} in parallel, and their combined output is added back to the residual stream before passing through the MLP, whose output is then added residually to form x_i.

Each attention head is parameterized by four matrices: query W_Q, key W_K, value W_V, and output W_O, which form the following composite matrices:

$$W_{QK} := W_Q^\top W_K, \quad W_{OV} := W_O W_V.$$

Using these matrices, along with the embedding matrix W_E and the unembedding matrix W_U, the attention computation for each head decomposes into two core circuits.

The *Query-Key (QK) circuit*, defined as $W_E^\top W_{QK} W_E$, provides the attention scores for every query–key token pair. Intuitively, each entry describes how much a given query token wants to attend to a given key token, providing insights *where* information flows within the model.

The *Output-Value (OV) circuit*, defined as $W_U W_{OV} W_E$, determines *what* information is transferred to the output logits when a token is attended to.

This formulation also allows transformers to be represented as a computational graph, where nodes correspond to components like attention heads or MLPs, and edges represent learned weights. Circuits, subgraphs specialized for particular tasks, can then be identified and studied mechanistically.

Indirect Object Identification Wang et al. (2022) analyzed GPT-2 small's performance on the IOI task, where the model must predict the indirect object (IO) in sentences like: "*When Mary and John went to the store, John gave a bottle of milk to Mary.*" The correct prediction is "*Mary*", not the repeated subject "*John*".

A human-interpretable strategy to solve IOI involves three steps: (1) identify all names in the sentence, (2) remove duplicates, and (3) output the remaining name. GPT-2 small mirrors this algorithm through three distinct attention head groups: Duplicate Token Heads detect repeated names, attending from the second mention back to the first; S-Inhibition Heads suppress repeated tokens; and Name Mover Heads copy the correct IO into the output via attention.

Path Patching Path patching is an intervention-based interpretability method for circuit discovery (Wang et al., 2022). It utilizes two prompt distributions: the original task distribution p_{orig}, and a corrupted distribution p_{new} designed to break task-relevant behavior. First, the model is run on

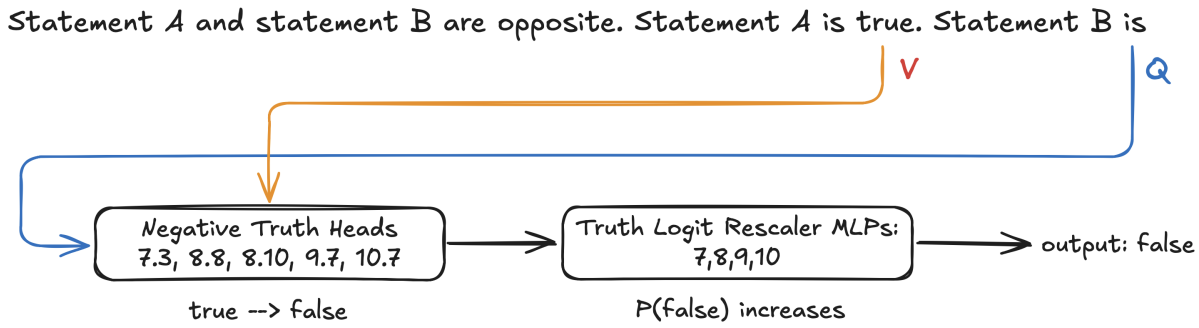


Figure 1: Opposite Syllogism Circuit. The blue arrow represents queries, and the yellow arrow represents values. Negative Truth Heads perform negation of the truth value present in the prompt while the Truth Logit Rescaler MLPs rescale the residual stream to select the correct truth value.

both distributions and each computational node's activations are cached. Then, a forward pass is performed on p_{orig} where the activation at a specific node F (e.g., an attention head) is replaced with its counterpart from p_{new}, while the rest of the model remains unchanged. Next, the resulting activation at a downstream node G is cached and patched into a forward pass on p_{new}. The causal impact of the path $F \to G$ is quantified by measuring the change in logit difference. A substantial drop indicates that F is causally important for the model's behavior on the task.

Logit Lens The logit lens (Nostalgebraist, 2020) is an interpretability method that projects the hidden state of a computational node, h, into the model's token space. It applies layer normalization followed by the unembedding matrix:

$$\text{LogitLens}(h) = \text{LayerNorm}(h)W_U.$$

This yields a distribution over tokens, revealing which outputs the model would favor if it predicted directly from that point.

3 Understand How GPT-2 Process Syllogisms

Syllogisms (Aristotle, c. 350 BC) offer an effective way to analyze a LM's reasoning capacity. Rather than analyzing a broad range of facts in a syllogistic format, we narrow our focus to a simpler set of propositions and declarative statements such as: "*Statement A is true. Statement B matches statement A. Statement B is true.*" We define three types of syllogisms: **Simple Syllogism (SS)**, **Opposite Syllogism (OS)**, and **Complex Syllogism (CS)**. A complete example of each type is provided in Table 1. We define the logit difference for the syllogism task family as follows. Let the answer set be $S = \{\text{true}, \text{false}\}$ with the correct answer $x \in S$, and the incorrect answer $\neg x \in S$. The logit difference (LD) is then given by:

$$\text{LD} = \text{logit}(x) - \text{logit}(\neg x).$$

A positive logit difference indicates that the first logit is more probable, while a negative logit difference suggests the second logit is more probable. e^{LD} represents how many times more likely the model will predict x compared to $\neg x$. Thus, for the SS format, GPT-2 small is 6.4077 times more likely to predict the correct truth value.

To quantify how well a circuit preserves model behavior, we use the **faithfulness** metric. Let $LD(\mathcal{M})$ denote the average logit difference (ALD) of the full model $\mathcal{M}$, and $LD(\mathcal{C})$ that of a circuit $\mathcal{C}$. The faithfulness metric is defined as:

$$\text{Faithfulness} = |ALD(\mathcal{M}) - ALD(\mathcal{C})|.$$

A lower value indicates that the circuit faithfully recovers the model's behavior on the task.

3.1 Simple Syllogism

We frame the SS task with the following human-interpretable algorithm: (1) Identify the single truth value token in the prompt; (2) Output the truth value token. Construction of the SS dataset can be found in Appendix A.

Truth Heads We begin by applying path patching to determine which attention heads and MLPs influence the model's output logits on SS prompts. As shown in Figure 2a, MLP layers have minimal direct effect on the logits, suggesting they are not essential for solving the SS task. We explore this further in Appendix C.

In contrast, Figure 2b reveals that several attention heads in the later layers, particularly heads 7.2, 9.1, 9.9, 10.1, and 10.4, contribute substantially to logit differences. To understand the behavior of these heads, we analyze their attention patterns using their QK circuits. Specifically, for each attention head h, we compute the raw attention score:

$$A^h = t^\top W_E^\top W_{QK}^h W_E t,$$

which captures how much each query token attends to each key token in the vocabulary space.

Type	Example Syllogism	Avg. Logit Diff.
Simple	*Statement A is true. Statement B matches statement A. Statement B is true*	1.8575
Opposite	*Statement A and statement B are opposites. Statement A is true. Statement B is false*	1.2123
Complex	*Statement A is true. Statement B matches statement A. Statement C is false. Statement B is true*	1.3105

Table 1: Examples of syllogism types with their corresponding average logit differences over datasets of 500 prompts. The LM is expected to predict the red tokens. We create these distributions of syllogism by replacing letters and truth values.

We find that these heads exhibit similar induction-like attention patterns: they predominantly attend to the final token corresponding to the truth value.

We provide a visualization of the most influential head, 7.2, in Figure 2c, along with the top $K = 3$ query–key token pairs in Table 2. For reference, we use the example SS prompt: "*Statement E is true. Statement S matches statement E. Statement S is true*". Across top heads we consistently observe two high-scoring token pairs: (`S`, `matches`) and (`is`, `true`). The first pair indicates that GPT-2 has developed a logical understanding of equivalence between the two statements—effectively computing $matches(S, E)$—while the second pair shows it retrieving the correct truth value based on this relationship. This consistent behavior leads us to call these attention heads *Truth Heads*.

To test whether Truth Heads depend on earlier attention heads, we repeat path patching on their query, key, and value inputs. We find that earlier heads have minimal effect, suggesting that the Truth Heads operate independently. To verify their sufficiency we build a minimal circuit C_{SS} consisting only of the Truth Heads. C_{SS} faithfully recovers the predictions of the model, achieving an average logit difference of 1.9286, effectively matching the performance of the GPT-2 small on the task. The Truth Heads' QK circuit consistently directs attention to the correct truth value earlier in the prompt while their OV circuit copies that value into the residual stream at the final token. Using the logit lens on truth heads confirms their output strongly favors the correct truth value.

Negative Heads in Simple Syllogism In addition to the Truth Heads, we identify a distinct group of heads—such as 9.7, 10.7, and 11.10—that exhibit attention patterns similar to the Truth Heads but are not essential for solving the task. Notably, mean-ablating head 10.7 improves model performance beyond the baseline. Head 10.7 has previously been characterized as a negative head in prior work (Wang et al., 2022) where it was shown to reduce the logit of specific output tokens. We hypothesize these negative heads encode the logit of the incorrect class in a binary setting. This aligns with findings from the copy suppression literature (McDougall et al., 2023) where head 10.7 was also found to suppress certain tokens. To test this, we turn to the Opposite Syllogism format.

3.2 Opposite Syllogism

To test our hypothesis surrounding negative heads, we investigate the model's behavior on *opposite syllogisms* (OS). We define a human-interpretable algorithm for this task in three steps: (1) Identify the single truth value token in the prompt, (2) Negate the truth value token, and (3) Output the negated token. Details on dataset construction are provided in Appendix A. For reference, we use the example OS prompt: "*Statement E and statement S are opposites. Statement E is true. Statement S is false*".

Negative Truth Heads. We begin by identifying components that directly influence the model's output on OS prompts. Path patching shows that ablating attention heads 7.3, 8.8, 8.10, 9.7, and 10.7 leads to a significant drop in logit difference (Figure 3b). To understand their role, we analyze each head's output by applying its OV matrix to the MLP-extended embedding basis, following prior techniques from Wang et al. (2022) and McDougall et al. (2023):

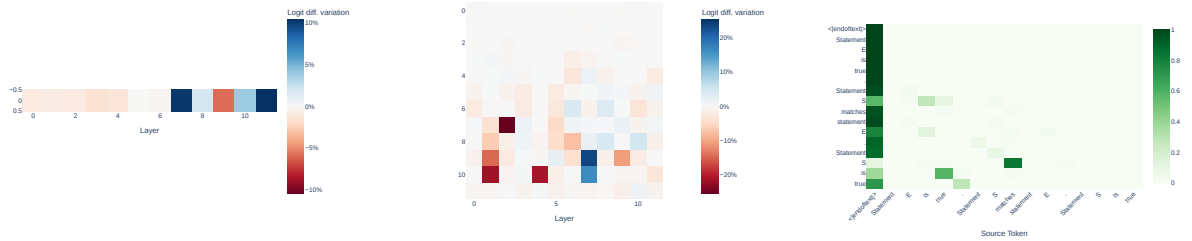
$$W_U \, W_{OV}^h \, \mathrm{MLP}_0(W_E).$$

These heads consistently attend to the truth value token in the prompt (e.g., `is`, `true`)—mirroring behavior observed in the SS setting. However, their influence on the logits differs. Some heads, like 8.8 and 5.1, promote the truth token to the top logits and function as standard *Truth Heads*. In contrast, heads such as 7.3, 8.10, 9.7, 10.7, and 11.10 suppress the truth token into the bottom logits (Table 3). We refer to this group as *Negative Truth Heads*.

Unlike the SS format, MLPs corresponding to these heads are crucial: ablating them significantly reduces performance (Figure 3a). Furthermore, path patching the queries of these Negative Truth Heads confirms they operate independently, with

Head	1st Highest Q–K Pair	2nd Highest	3rd Highest
7.2	**0.8304:** `['S', 'matches']`	**0.5749:** `['is', 'true']`	0.2750: `['true', '.']`
10.1	**0.7139:** `['is', 'true']`	**0.5524:** `['S', 'matches']`	0.4258: `['S', 'true']`
10.4	**0.6833:** `['is', 'true']`	**0.6637:** `['S', 'matches']`	0.5063: `['.', 'Statement']`

Table 2: Top 3 highest-scoring query–key token pairs from the attention pattern scores of the most influential heads in the SS format.



(a) MLP Direct Effect on Logits (b) Attention Head Direct Effect on Logits (c) QK Circuit Visualization (Head 7.2)

Figure 2: Path Patching and QK Analysis on Simple Syllogism Prompts.

no significant upstream influence—mirroring earlier findings from SS where only components with direct logit impact matter.

Mechanistic Interpretation The QK circuit of each Negative Truth Head reliably identifies the truth token—fulfilling Step 1 of the OS algorithm. However, their OV projection suppresses this token into the bottom logits. The associated MLP then rescales the residual stream to promote the opposite of the suppressed token into the top logits, completing Steps 2 and 3. We refer to these components as *Truth Logit Rescaler MLPs*.

This attention–MLP sequence forms a mechanistic pathway for learned negation: the model suppresses a truth token and then elevates its negation for output. Table 3 captures the entire trajectory of such a token: from attention-induced suppression to MLP-driven recovery. This reveals how the model predicts a correct token not seen in the prompt, using the suppression of the incorrect token as a signal for its opposite. While this attention–MLP mechanism effectively negates the truth value in opposite syllogisms, we observe a consistent asymmetry: the negation process is more reliably triggered when the input token is `true`, resulting in `false` predictions. In contrast, when the prompt contains `false`, the model often retains `false` as the dominant logit rather than flipping to `true`.

Circuit Faithfulness To test sufficiency, we construct a circuit C_{OS} using only the Negative Truth Heads and their associated MLPs. This circuit recovers approximately 85% of GPT-2 small's performance, demonstrating it is a faithful subcircuit for solving the OS task. A schematic of this circuit is shown in Figure 1.

Reversal of Head Behavior Between Tasks Interestingly, the same heads that negatively affected logit difference in the SS task—like 10.7—now play a constructive role in OS. This reversal demonstrates that the model reuses certain components in complementary tasks where their function flips to support inverse outcomes. We extend this finding in Appendix B, where we test circuit generalization and transferability across other binary pairs beyond `true`/`false`, further supporting the idea that GPT-2 small represents logical negation via attention and MLPs.

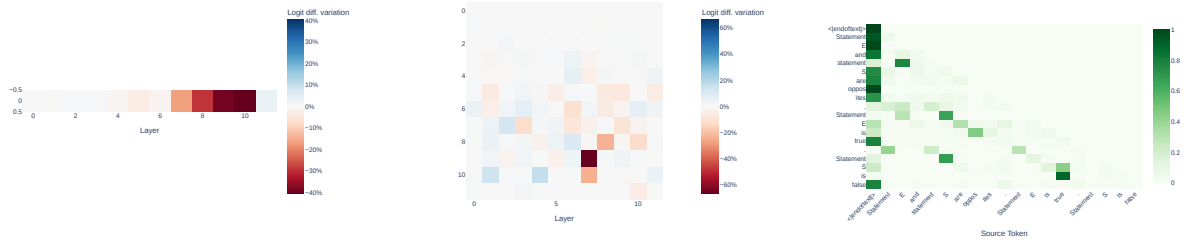
3.3 Complex Syllogism

The complex syllogism task expands on the previous setups by introducing a misleading, redundant statement. The objective is to determine whether GPT-2 Small can still arrive at the correct conclusion in the presence of potentially confusing information.

We begin with path patching to identify which attention heads and MLPs directly influence the model's output logits. The results are similar to the path patching results of the OS format. More specifically, MLPs in layers 8,9, and 10 positively influence the logit difference. Similar heads—specifically, heads 7.3, 8.8, 8.10, 9.7, and 10.7— were found to positively influence logit difference but with differ-

Top QK Pairs (Head 10.7)		
0.892: (‘is’, **‘true’**), **0.772:** (‘statement’, ‘E’), **0.685:** (‘Statement’, ‘S’), **0.662:** (‘Statement’, ‘S’), **0.459:** (‘is’, ‘oppos’)		
Stage	**Top Logits**	**Bottom Logits**
After OV from Head 10.7	depot, rink, carp, Dj, Hack, DJ, Gaz, Phillips, District, TTC	**‘true’**, ‘True’, ‘TRUE’, **‘true’**, ‘untrue’, ‘Null’
After MLP Layer 10	**‘true’**, **‘false’**, **‘True’**, **‘False’**, infinite, truly	blitz, ombo, plateau, corrid, tradem, emale, Citiz, sugg

Table 3: Top QK pairs in Head 10.7 strongly attend to truth-related tokens (e.g., ‘is’, ‘true’). Initially, the OV output does not rank truth tokens highly. However, after the MLP layer, both **‘true’** and **‘false’** become top-ranked, indicating the MLP can help produce the opposite token, even when it is not in the prompt



(a) MLP Direct Effect on Logits (b) Attention Head Direct Effect on Logits (c) QK Circuit Visualization (Head 10.7)

Figure 3: Path Patching and QK Analysis on Opposite Syllogism Prompts.

ent behavior. Heads 9.7, 10.7, and 11.10 attend to both truth tokens in the prompt but place greater emphasis on the incorrect (redundant) truth token. Inspecting their logits, we find these heads continue to perform a suppression operation as observed previously, and we therefore classify them as *Negative Truth Heads* in the CS format and do not investigate them further. In contrast, heads 7.3, 8.8, and 8.10 exclusively attend to the incorrect truth value, and we study them more closely to understand their contribution. We refer to these as *Truth Modulation Heads*.

To interpret the behavior of the Truth Modulation Heads, we analyze how their outputs project onto the unembedding vectors of the truth tokens. Let $W_U[\texttt{true}]$ and $W_U[\texttt{false}]$ represent the unembedding directions for the correct and incorrect truth tokens respectively. For a head output $h_i(X)$, we compute the logit contribution to token $t \in \{\texttt{true}, \texttt{false}\}$ as

$$\langle h_i(X), W_U[t] \rangle.$$

This dot product reflects how strongly head h_i pushes the residual stream toward generating token t. We scatter plot each head’s attention probability on the truth value token against the corresponding logit contribution along the true direction. Two distinct patterns emerge across the heads. First, the correct token is consistently ignored by these heads, receiving very low attention probability. Second, the incorrect token shows two opposing trends depending on the head: in some cases, the attention probability and logit contribution are positively correlated, suggesting that the head reinforces the incorrect truth value. We refer to such heads as *Correct Truth Inhibition Heads*. In other cases, the relationship is negative—the more attention the head gives to the incorrect token, the more it pushes away from the incorrect truth direction. This effectively reinforces the correct token and we refer to these as *Correct Truth Reinforcement Heads*.

These behaviors are further supported by examining the top and bottom logits. In inhibition heads, the incorrect token consistently appears among the top logits while in reinforcement heads it appears among the bottom logits. This supports the interpretation that Truth Modulation Heads implement a binary operation: either reinforcing or inhibiting the direction of the incorrect token, which indirectly determines the correct output.

We then investigate whether the Negative Truth

Heads and Truth Modulation Heads influence one another. Path patching reveals that neither group affects the other directly, although both are influenced by similar upstream sources. Moreover, ablating one group does not destroy faithfulness, confirming that the groups can operate independently. This redundancy is consistent with the findings of McGrath et al. (2023), who describe the Hydra Effect in language models, where multiple pathways can implement the same behavior.

Path patching reveals that both the Negative Truth Heads and the Truth Modulation Heads receive input from a shared set of upstream heads, primarily located in Layers 0 and 5. In Layer 5, heads 5.1 and 5.5 exhibit classic induction patterns as described by Elhage et al. (2021), attending from the conclusion line (e.g., "*Statement B is*") back to a matching premise. This effectively links the conclusion to its logical source. In Layer 0, other heads attend to repeated statement identifiers (e.g., "*Statement B*") and influence both the key vectors of downstream heads and the values used by the Induction Heads. These heads appear to detect repeated statements and function as *Duplicate Statement Identifier Heads*, marking the reuse of information—an essential step in a natural deduction process.

At inference time, the Induction Heads serve as a routing mechanism: they either connect the conclusion to the premise via the "*matches*" relation or directly extract the incorrect truth value from the conclusion line. This information is then processed by the Negative Truth Heads or Truth Modulation Heads to generate the final token.

4 Discussion

Connection to IOI Despite overlap with the Name Mover Heads from the IOI task, we find (Negative) Truth Heads reflect broader functionality, particularly negate rather than simply copy. From an IOI perspective, the Negative Truth Heads were initially interpreted as negative copy heads due to their tendency to replicate the tokens they attend to. However, in the opposite syllogism task, the correct answer is not explicitly present in the prompt. Consequently, these heads cannot simply copy the attended truth value to produce the correct answer. This provides strong evidence that Negative Truth Heads encode the direction of the less contextualized logit in a binary setting, effectively operating in the antidirection. We believe this behavior remained unnoticed in IOI because, in that context, Mary $\neq \neg$John. Similarly, many of the Truth Modulation Heads align with the S-Inhibition category from IOI, suggesting a shared functional role. We identify the Correct Truth Inhibition Heads as the original inhibition heads from IOI, given their role in reinforcing focus on the incorrect token. This expanded understanding highlights how heads previously characterized in IOI tasks can exhibit more nuanced and adaptable behaviors in different contexts.

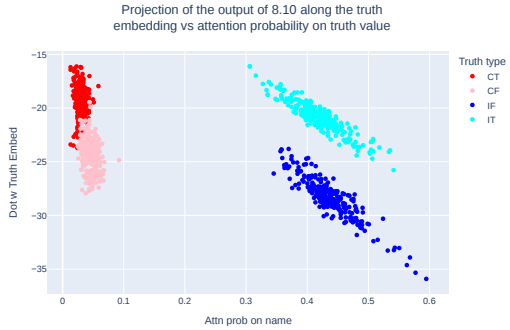
Clustering of Truth Modulation Heads We observe distinct clusters in both groups of Modulation Heads. To refine our truth types, we categorize truth values into four types: correct true (CT), correct false (CF), incorrect true (IT), and incorrect false (IF). This results in two natural pairings: (CT, IF) and (CF, IT). As shown in Figure 4a and 4b, `false` (IF or CF) has larger projections on the truth embedding. We believe that this asymmetrynot only reflects the internal bias of truth values learned from the training ccorpus, but also resembles the behavior observed in the Opposite Syllogism task, where negation was easier with `true`. Although we do not rigorously analyze this connection, it may reflect a broader model bias toward negative truth values or a negation-like structure in its internal representations.

Scalability of Results We extend our simple and opposite syllogism formats to larger models such as GPT-2 XL, Pythia 1.4B, Qwen3-1.7B, and LLaMA3.2-1B. (See Appendix D).

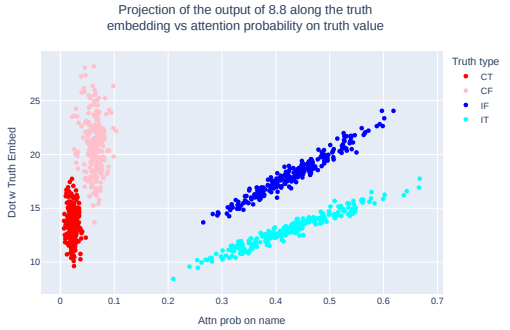
5 Related Works

Mechanistic Interpretability Mechanistic Interpretability research offers various techniques to reverse-engineer model behavior and identify important components responsible for a model's performance. In addition to Path Patching (Wang et al., 2022), there are other patching methods including Attribution Patching (Nanda, 2023), causal mediation analysis (Meng et al., 2023; Pearl, 2022; Vig et al., 2020), and AtP* (Kramár et al., 2024). Sparse Autoencoders (SAEs) have become increasingly popular for interpreting features (Bricken et al., 2023; Marks et al., 2025). Earlier works such as Neuron Shapley (Ghorbani and Zou, 2020) introduce a framework that quantifies each neuron's contribution to a deep network's performance by considering interactions among neurons. Other earlier works such as Cao et al. (2021); Csordás et al. (2021) employ subnetworks to investigate what model internals are needed to perform a task through probing and masking.

Circuit Discovery in GPT-2 IOI has inspired many other circuit analysis works. (Hanna et al., 2023) identify a circuit that explains GPT2's ability to predict correct year tokens when prompted with task like "The war lasted from the year 1732 to 17". (Merullo et al., 2024) rediscover the IOI circuit in GPT2-Medium and show that much of the circuit can be reused to solve the Colored Object task introduced by (Srivastava et al., 2023). (Nainani et al., 2024) explore IOI's generality by extending the prompt to include more instances



(a) Correct Truth Reinforcement Head 8.10 Projection along truth embedding versus attention probability on truth value



(b) Correct Truth Inhibition Head 8.8 Projection along truth embedding versus attention probability on truth value

Figure 4: The Truth Modulation group with refined truth types

of the indirect object. Conmy et al. (2023) generalize and automate the IOI-style analysis within GPT-2 small, ultimately recovering many already discovered circuits.

Syllogisms for Assessing LLMs Recent studies have explored assessing LLMs with syllogisms. Eisape et al. (2024); Ando et al. (2023) provide a comparative analysis on how humans and LLMs perform syllogistic reasoning. Kim et al. (2025) conduct a mechanistic analysis of standard syllogisms. In contrast, our work explores syllogisms with assigned truth values, offering a distinct perspective. Furthermore, we provide novel insights into the role of MLPs in facilitating syllogistic reasoning and handling logical negation.

6 Conclusion

In this work, we reverse-engineered GPT-2 for three syllogism tasks of varying complexity, uncovering key insights into how GPT-2 handles binary truth values within logical tasks. In the simplest case, high faithfulness was achieved with just Truth Heads, highlighting the model's ability to maintain correct truth values with minimal components. In the opposite syllogism case, the inclusion of Negative Truth Heads and MLPs allowed the model to properly negate the truth value, demonstrating the novel negation mechanism in handling binary outcomes. In the complex case, while negation remained a key mechanism, additional heads were needed to identify and process the correct truth value to negate, reflecting the increased complexity of the task. Our findings reveal significant overlap with the IOI circuit, expanding the understanding of these computational nodes' capabilities.

7 Limitations

Our study focuses on GPT-2 Small, a relatively early and compact language model. While this model is widely used in mechanistic interpretability research due to its manageable size, its age and scale may limit the generalizability of our findings to more modern and larger models. Although we perform limited extensions to larger models, a deeper analysis across model families and scales remains an open direction for future work. Additionally, our syllogism datasets are synthetic and templated, which may not reflect the full diversity and ambiguity of natural language logical reasoning.

8 Ethical Considerations

This work does not involve human subjects, private user data, or any personally identifiable information. Our experiments are conducted entirely on publicly available models and synthetic prompts. However, we acknowledge that large language models can encode and amplify societal biases present in their training data. While our work does not directly address such issues, future investigations into the interaction between logical circuits and bias propagation could be valuable. We also note that reverse-engineering model internals, while useful for scientific understanding, should be accompanied by considerations of responsible disclosure when vulnerabilities or harmful behaviors are uncovered.

Acknowledgments

We thank the broader mechanistic interpretability community for open-sourcing tools and models that made this work possible. In particular, our experiments and analyses are built upon `TransformerLens` (Nanda and Bloom, 2022), which provided essential infrastructure for probing and visualizing transformer internals. This research was not supported by any external funding.

References

Risako Ando, Takanobu Morishita, Hirohiko Abe, Koji Mineshima, and Mitsuhiro Okada. 2023. Evaluating large language models with neubaroco: Syllogistic reasoning ability and human-like biases. *Preprint*, arXiv:2306.12567.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Preprint*, arXiv:2406.11717.

Aristotle. c. 350 BC. *Prior Analytics*. Oxford University Press, Oxford, UK.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Steven Cao, Victor Sanh, and Alexander M. Rush. 2021. Low-complexity probing via finding subnetworks. *Preprint*, arXiv:2104.03514.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. *Preprint*, arXiv:2304.14997.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. 2021. Are neural nets modular? inspecting functional modularity through differentiable weight masks. *Preprint*, arXiv:2010.02066.

Tiwalayo Eisape, MH Tessler, Ishita Dasgupta, Fei Sha, Sjoerd van Steenkiste, and Tal Linzen. 2024. A systematic comparison of syllogistic reasoning in humans and language models. *Preprint*, arXiv:2311.00445.

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.

Amirata Ghorbani and James Zou. 2020. Neuron shapley: Discovering the responsible neurons. *Preprint*, arXiv:2002.09815.

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

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Preprint*, arXiv:2305.00586.

Geonhee Kim, Marco Valentino, and André Freitas. 2025. A mechanistic interpretation of syllogistic reasoning in auto-regressive language models. *Preprint*, arXiv:2408.08590.

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

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2025. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *Preprint*, arXiv:2403.19647.

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

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. 2023. The hydra effect: Emergent self-repair in language model computations. *Preprint*, arXiv:2307.15771.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023. Locating and editing factual associations in gpt. *Preprint*, arXiv:2202.05262.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024. Circuit component reuse across tasks in transformer language models. *Preprint*, arXiv:2310.08744.

Jatin Nainani, Sankaran Vaidyanathan, AJ Yeung, Kartik Gupta, and David Jensen. 2024. Adaptive circuit behavior and generalization in mechanistic interpretability. *Preprint*, arXiv:2411.16105.

Neel Nanda. 2023. Attribution patching: Activation patching at industrial scale.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. `https://github.com/TransformerLensOrg/TransformerLens`.

Nostalgebraist. 2020. Interpreting gpt: The logit lens.

Judea Pearl. 2022. *Direct and Indirect Effects*, 1 edition, page 373–392. Association for Computing Machinery, New York, NY, USA.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 432 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Preprint*, arXiv:2206.04615.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *Preprint*, arXiv:2211.00593.

A Syllogism Dataset Construction

Syllogistic prompts were created using templates. `[TRUTH_VALUE]` $\in$ {`true`, `false`}, and `[A], [B], [C]` are sampled from capital letters. See table 4 for templates of each syllogism format

B Generality Across Binary Contrasts

Having established mechanistic evidence for circuits supporting binary truth tasks in both the simple and opposite syllogism settings, we next evaluate the generality of these circuits beyond the original `true/false` framing. Specifically, we test whether the same circuits generalize to analogous binary pairs: `(right, wrong)`, `(good, bad)`, `(positive, negative)`, and `(correct, incorrect)`.

To do so, we apply both the simple syllogism circuit ($\boldsymbol{C}_{SS}$) and opposite syllogism circuit ($\boldsymbol{C}_{OS}$) to each pair and compare their performance to the full GPT-2 Small model. As shown by tables 5 and 6 we find that the original circuits often match or even outperform the full model in logit difference between most binary pairs of tokens. This provides compelling evidence that the binary task is not specific to a particular token pair, but instead reflects a transferable reasoning mechanism.

To further validate generalization, we visualize direct path patching attention results across each binary pair. As seen in Figures 5–8, across the binary pairs of tokens, the core attention heads relevant to the simple and opposite syllogism cases are opposite in their effect on logit difference.

C Disentangling MLP Contributions via Patching

To assess the contribution of MLPs to the model's output, we perform path patching both with and without attention restored. Figure 10b shows that early-layer MLPs—particularly MLP0—appear to significantly affect the logits when patched in isolation. This aligns with prior observations that MLP0 functions as an extended embedding layer, especially when attention is absent (McDougall et al., 2023; Wang et al., 2022).

However, once attention is also restored, the influence of these early MLPs sharply diminishes. This suggests their apparent impact in the no-attention condition is largely an artifact of missing context, rather than a reflection of GPT2 semantic ability to complete syllogisms.

For this reason, in all subsequent experiments analyzing MLP effects, we report results with attention paths patched in. This allows us to isolate the true downstream influence of MLPs under more realistic model conditions.

D Extension to Larger Models

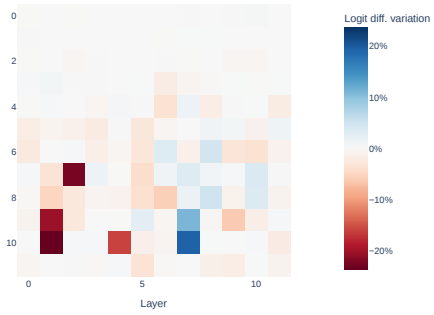
To assess whether the findings observed in GPT-2 Small generalize across model scale and architecture, we extend our experiments to several larger models: GPT-2 XL, Pythia 1.4B, Qwen3-1.7B, and LLaMA3.2-1B.

Across all models, we continue to observe empirical signatures of binary behavior: heads relevant to the simple and opposite syllogism tasks tend to exert opposing effects on the logits. MLP layers remain important in the opposite syllogism task for all models except Pythia 1.4B, mirroring the behavior observed in GPT-2 Small. Notably, Table 7 shows that performance on the simple syllogism format degrades significantly in larger models, suggesting that task generalization does not uniformly scale with model size.

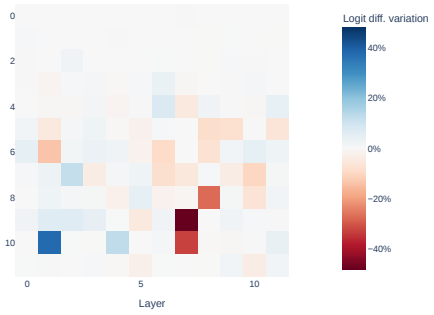
All models retain some attention heads exhibiting negative-copy behavior. However, the influence of these heads on output logits is more muted compared to GPT-2 Small. In particular, the heads most responsible for enabling opposite syllogism performance in the larger models are not the negative heads. Qwen3-1.7B, for instance, contains relatively few negative heads, and those it has do not drive logit differences in either task. An exception is Pythia 1.4B, whose success on the opposite task remains closely tied to the activity of its negative-copy heads.

Interestingly, across all models, the heads most influential on model output tend to exhibit strong induction behavior (e.g., ABA $\rightarrow$ B), regardless of whether they also contribute to the task-relevant distinction. Yet despite this variability in attention head dynamics, the consistent involvement of MLPs in the opposite task—and their near absence in the simple task—suggests a robust division of labor: negation appears to depend more heavily on the feedforward path than on attention alone. This may help constrain future hypotheses about the mechanistic implementation of logical inversion and contextual negation in transformer models.

These findings remain empirical and exploratory. Figures 11-14 illustrate the direct effects of attention heads and MLPs across the syllogism tasks. A deeper investigation into how architectural scale affects circuit behavior remains a promising direction for follow-up work.

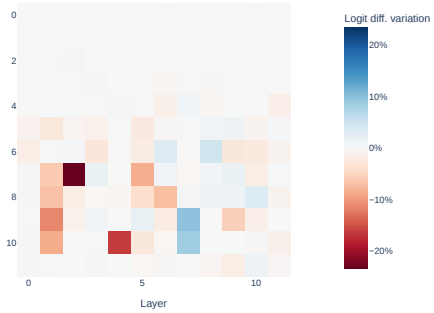


(a) Simple Syllogism with Right/Wrong

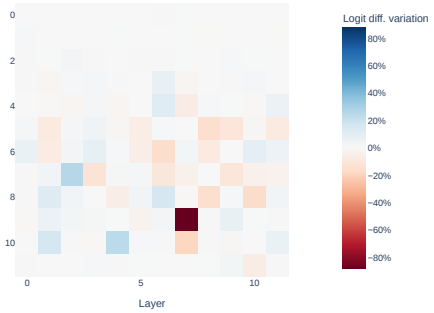


(b) Opposite Syllogism with Right/Wrong

Figure 5: Binary task results of Right/Wrong

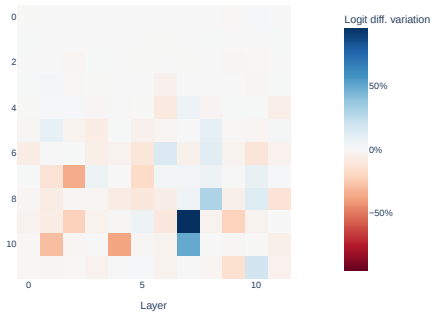


(a) Simple Syllogism with Correct/Incorrect

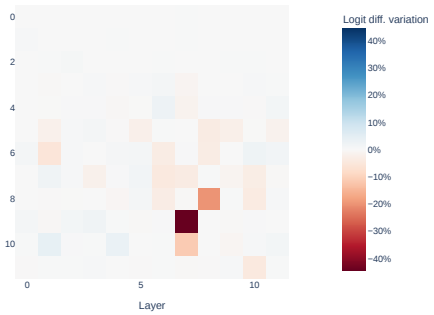


(b) Opposite Syllogism with Correct/Incorrect

Figure 6: Binary task results of Correct/Incorrect



(a) Simple Syllogism with Positive/Negative



(b) Opposite Syllogism with Positive/Negative

Figure 7: Binary task results of Positive/Negative

Type	Template
Simple Syllogism	1. Statement [A] is [TRUTH_VALUE_1]. Statement [B] has the same truth value as [A]. Statement [B] is [TRUTH_VALUE_1].
	2. Statement [A] is [TRUTH_VALUE_1]. Statement [B] matches statement A. Statement B is [TRUTH_VALUE_1].
	3. *(Extended)* Statement [A] is [TRUTH_VALUE_1]. Statement [B] must match [A]. Statement [C] doesn't matter. Statement [B] is [TRUTH_VALUE_1].
Opposite Syllogism	1. Statement [B] has the opposite truth value of [A]. Statement [A] is [TRUTH_VALUE_1]. Statement [B] is [TRUTH_VALUE_2].
	2. Statement [A] and statement [B] are opposites. Statement [A] is [TRUTH_VALUE_1]. Statement [B] is [TRUTH_VALUE_2].
Complex Syllogism	1. Statement [A] is [TRUTH_VALUE_1]. Statement [B] has same truth value as [A]. Statement [C] is [TRUTH_VALUE_2]. Statement [B] is [TRUTH_VALUE_3].
	(Harder constraint): [TRUTH_VALUE_2] = ¬[TRUTH_VALUE_1].
Complex Opposite Syllogism	1. Statement [A] is [TRUTH_VALUE_1]. Statement [B] has the opposite truth value of [A]. Statement [C] is [TRUTH_VALUE_2]. Statement [B] is [TRUTH_VALUE_3].
	2. Statement [A] and [B] are opposites. Statement [C] has the same truth value as [A]. Statement [B] is [TRUTH_VALUE_3].
	3. Statement [A] is [TRUTH_VALUE_1]. Statement [A] and [B] are opposites. Statement [C] is [TRUTH_VALUE_2]. Statement [B] is [TRUTH_VALUE_3].

Table 4: Templates used for generating syllogistic prompts.

	Original	Good/Bad	Pos/Neg	Correct/Incorrect	Right/Wrong
GPT-2 Small	1.8399	1.7738	0.6958	2.1221	2.0309
$\boldsymbol{C}_{SS}$	1.9234	1.9940	1.1584	1.6785	2.1599

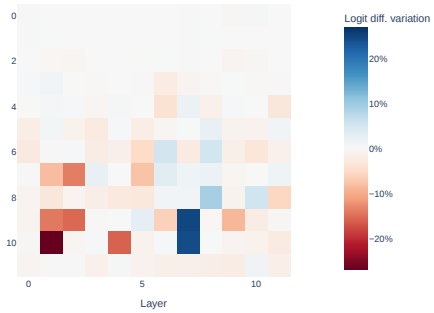
Table 5: Transferability of $\boldsymbol{C}_{SS}$ to other binary token pairs

	Original	Good/Bad	Pos/Neg	Correct/Incorrect	Right/Wrong
GPT-2 Small	1.2632	2.1163	3.0032	0.7986	1.3469
$\boldsymbol{C}_{OS}$	1.3136	1.7136	1.0113	0.8142	1.2481

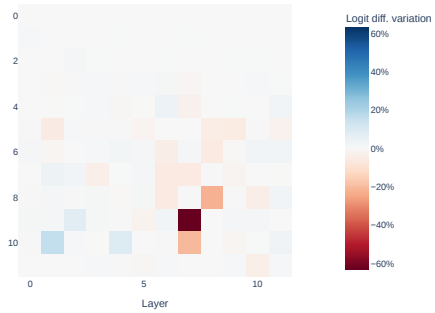
Table 6: Transferability of $\boldsymbol{C}_{OS}$ to other binary token pairs

	GPT-2 XL	Qwen3-1.7B	LLaMA 3.2-1B	Pythia 1.4B
Simple Syllogism	0.1112	0.5322	−0.4357	1.0105
Opposite Syllogism	2.6114	1.5257	−0.1807	2.1098

Table 7: Average logit difference across models and tasks.

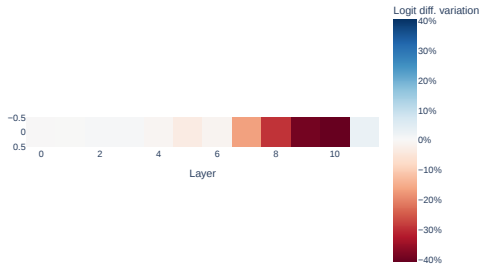


(a) Simple Syllogism with Good/Bad

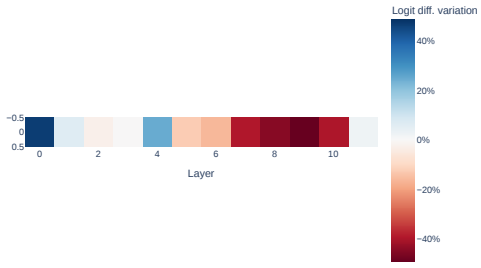


(b) Opposite Syllogism with Good/Bad

Figure 8: Binary task results of Good/Bad

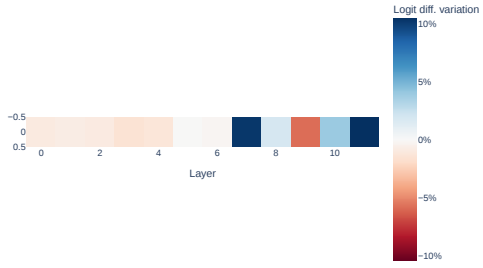


(a) MLP effects with attention context

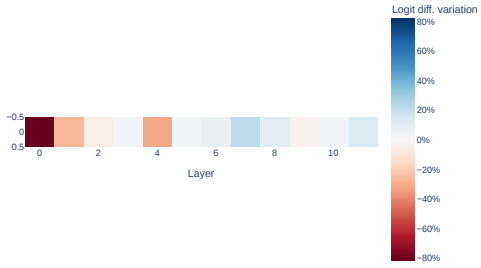


(b) MLP effects without attention context

Figure 9: Path patching MLPs in the opposite syllogism task. (a) shows effects when MLPs are patched with attention context preserved; (b) shows isolated MLP contributions without attention context.



(a) MLP effects with attention context



(b) MLP effects without attention context

Figure 10: Path patching MLPs in the simple syllogism task. (a) shows effects when MLPs are patched with attention context preserved. No MLPs have significant importance; (b) shows isolated MLP contributions without attention context. Early MLPs, specifically MLP0, appear relevant for the task

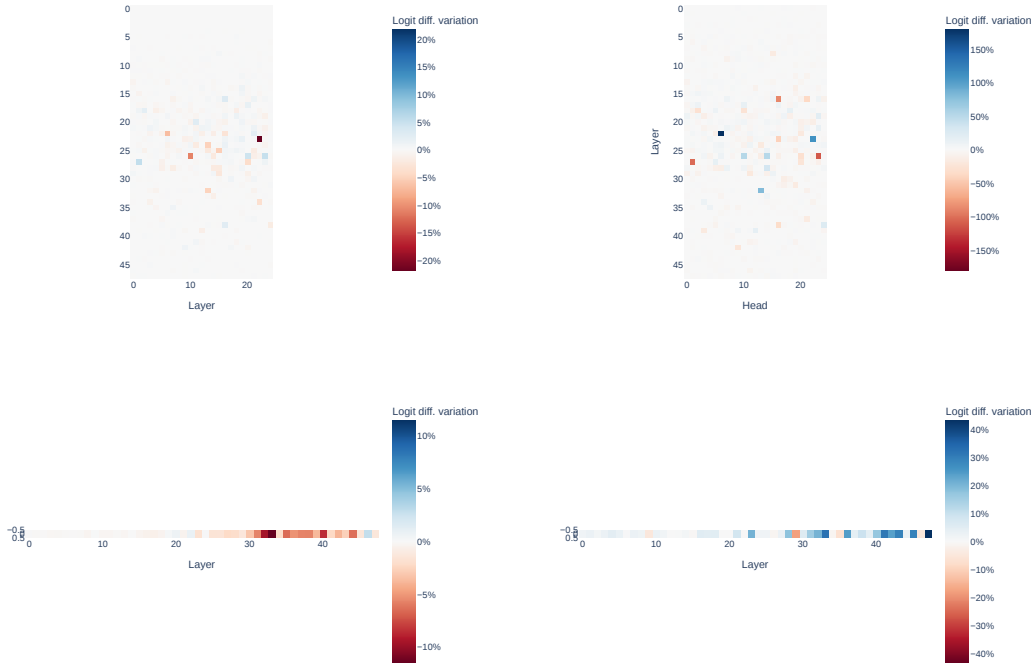


Figure 11: Direct effects of attention heads and MLPs for GPT-2 XL across syllogism tasks.

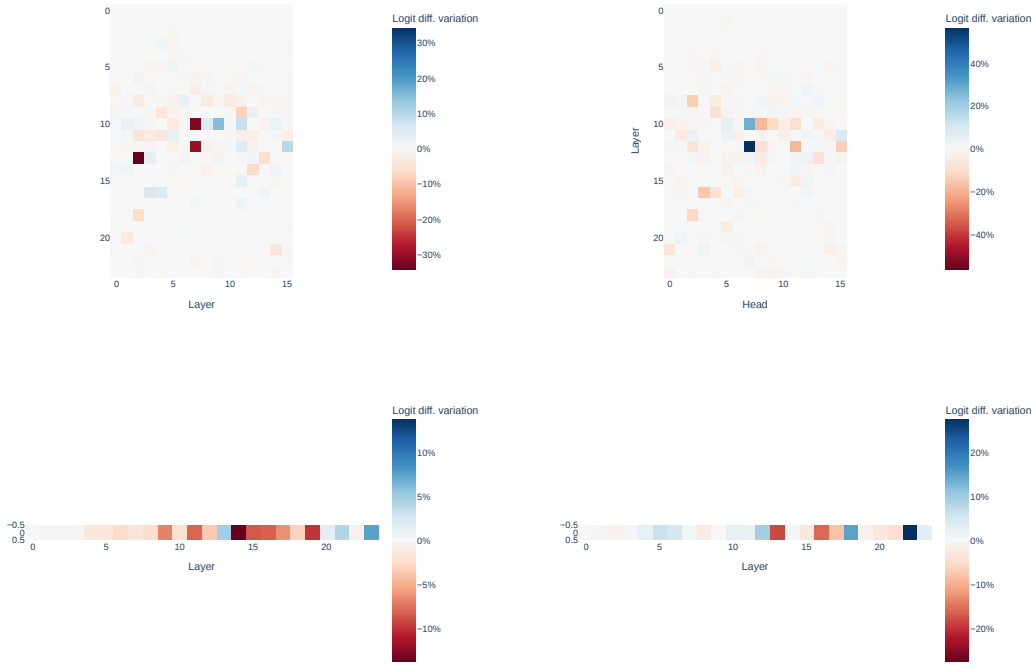


Figure 12: Direct effects of attention heads and MLPs for Pythia 1.4B across syllogism tasks.

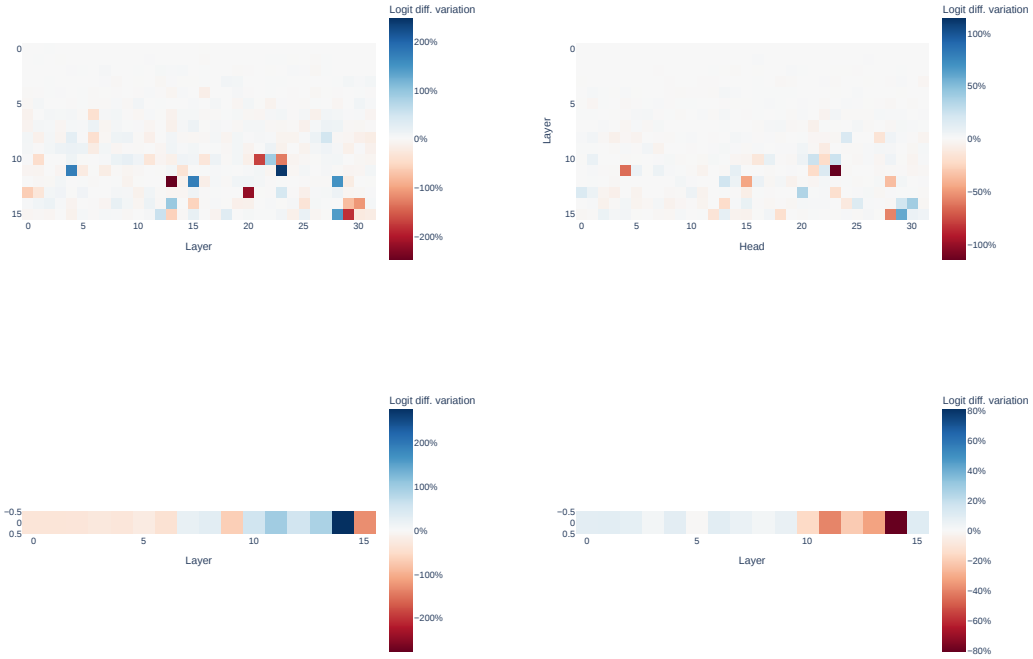


Figure 13: Direct effects of attention heads and MLPs for LLaMA 3.2B across syllogism tasks.

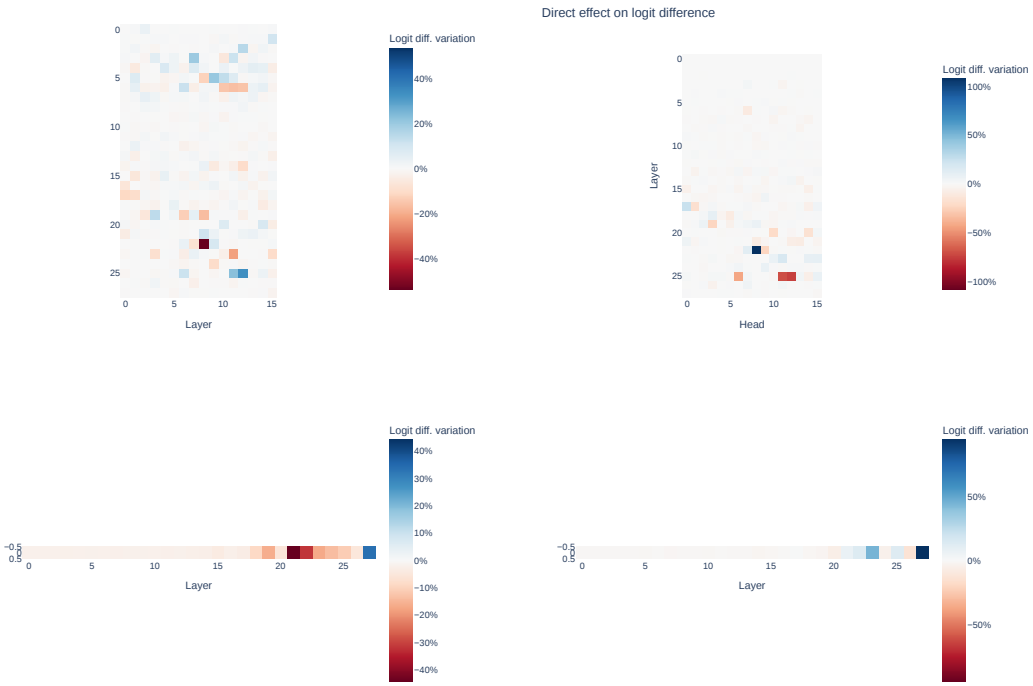


Figure 14: Direct effects of attention heads and MLPs for Qwen 1.7B across syllogism tasks.