For What Reason?
Interpreting Models' Encoding of Causation and Antithesis

Abhidip Bhattacharyya
University of Massachusetts Amherst
abhidipbhatt@umass.edu

Shira Wein
University of South Florida
shirawein@usf.edu

Abstract

Discourse relations provide document structure, critical to language understanding and enabling language model performance and ethicality. In this work, we investigate how instruction-tuned Transformer models (LLaMA and Mistral) encode discourse relations in English, with a particular focus on the contrasting relations of causation and antithesis. Framing the task as a next-token prediction task and applying a suite of interpretability techniques to test model internals, our findings show that certain early layers make predictive decisions at mid-sequence tokens, while some mid-level layers finalize their decisions closer to the last token. Most of the remaining layers primarily propagate earlier decisions rather than actively influencing them. Additionally, we observe that some layers exhibit a preference for one answer over alternatives, suggesting asymmetric representation of discourse-based reasoning.[1]

1 Introduction

Discourse relations are essential for natural language understanding, characterizing the structure of sentences within a document. Additionally, discourse structure is a critical component of enabling language model alignment with human preferences and promoting ethical behavior (Harrison et al., 2019; Hüsünbeyi et al., 2022; Adewoyin et al., 2022; Kim et al., 2026; Nakshatri et al., 2025). However, limited prior work has investigated discourse encoding within Transformers, with only one recent work has investigated discourse encoding in Transformers using Transformer circuit discovery to identify "discursive circuits" (Miao and Kan, 2025).

In this work, we investigate Transformer encoding of discourse-level phenomena in English with a focus on two frequent, and oppositional relations:

Causation: John studied hard, so he passed the exam.
Antithesis: John studied hard, yet he failed the exam.

(a) Examples illustrating causal and antithetical relationships. Causation links an action to its expected outcome, whereas antithesis presents a contrasting outcome.

Causation: John studied hard, *so* he was *able* to pass the exam.
Antithesis: John studied hard, *yet* he was *unable* to pass the exam.

(b) The same examples rewritten using the lexical contrast between *able* and *unable*. The conjunction "so" signals causation, while "yet" marks an antithetical relationship. Apart from the four highlighted tokens, the two sentences are identical.

causation and antithesis. In Rhetorical Structure Theory (RST; Mann and Thompson, 1987; Mann et al., 1989), causation represents the relationship between an action and its consequence. In contrast, the rhetorical figure of antithesis employs parallel grammatical structures to juxtapose contrasting or opposing ideas. An illustrative example of these discourse relations is shown in Figure 1a. Within our investigation of discourse encoding within Transformer models, our research questions include:

RQ1. Which token-level features drive the model's decisions between antithesis and causation?

RQ2. Which layers play a pivotal role in identifying antithesis and causation?

RQ3. What key concepts help the model distinguish between antithesis and causation?

RQ4. How does discursive information propagate across layers within the model?

[1]Our code is available at https://github.com/abhidipbhattacharyya/causation_vs_antithesis

Proceedings of the 27th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 237–253
August 2–5, 2026.

To address these research questions, like Miao and Kan (2025), we formulate the task as a next token prediction task, where the token indicates either a causal or antithetical relationship identified within the sentence. We perform a range of causal interventions and apply techniques from automated circuit discovery. Our results show that early and mid layers encode local relational meanings signaled by discourse markers such as "so"/ "yet," while higher layers primarily propagate or refine these signals. In contrast, interventions at the last token indicate that lower layers integrate sentence meaning, whereas higher layers make and maintain the final decision. We also observe that certain layers exhibit an inherent bias toward one option over the other. At the circuit level, we find a core set of stable connections that consistently drive decision making and information flow, although specific edges involved may vary depending on verb type and the number of demonstrations. Our findings provide insight into how critical discourse relations are encoded in Transformer models, enabling heightened model performance and alignment to human preferences.

2 Methods & Experiments

In this section, we detail the task formalism (Section 2.1), data preparation procedures (Section 2.2), model configurations (Section 2.3), and interpretability techniques (Section 2.4).

2.1 Task

We prompt models to complete the answer in a way that indicates a causation or antithesis within the sentence. To restrict the model prediction space we design the answer to be either "able" or "unable." We focus on decoder-only models, namely LLaMA 3 (Dubey et al., 2024) and Mistral (Jiang et al., 2023), using their instruction-tuned variants with 7 billion parameters. The task is formulated as a next-token prediction problem, allowing us to probe how these models internally represent and differentiate between distinct discourse relations.

We formulate the task of differentiating between causation and antithesis as a next-token prediction problem. In our setup, each sentence is constructed to include either "able" or "unable." For example, the sentences shown in Figure 1a are reformulated as illustrated in Figure 1b.

To prompt the LLM for this task, we replace 'able" or unable" with a blank and ask the model to predict the missing word, instructing it to choose only between able" and unable." The prompt includes few-shot examples (two per class) to clarify the task. We initially find with 1–6 shot prompting and found that 4-shot prompting produced more consistent results. Therefore, experiments in this paper follows 4 shots ((two per class). However, subsequent experiments show that 0-shot prompting achieved the best overall performance. Figure 11 in Appendix A presents prediction accuracy across different numbers of demonstrations, and an example prompt is shown in Figure 2. Additional details on prompt design are provided in Appendix A.

User Prompt: 1. Sentence: John got the train this morning, so he was ___ to reach on time.
Answer: able
2. Sentence: The team practiced all week, so they were ___ to perform well.
Answer: able
3. Sentence: The team practiced all week, yet they were ___ to perform well.
Answer: unable
4. Sentence: Lena studied hard, yet she was ___ to pass the exam.
Answer: unable

Now complete the following:
5. Sentence: He slept well last night, yet he was ___ to run this morning.
Answer:

Figure 2: Example of our few-shot prompt. Our system prompt can be found in Figure 10 in Appendix A.

2.2 Data

To probe the models' representation of causation and antithesis, we build a small dataset of sentences containing antithesis and their corresponding causation relations. We impose a constraint that both sentences in each pair must have the same word count. This constraint is motivated by our planned causal intervention experiments (Section 2.4).

To generate the dataset, we prompt ChatGPT 4o[2], by first manually creating a few example pairs that illustrate the contrast between causation and antithesis. These examples are then used to prompt

[2]Accessed May 2025.

ChatGPT to generate additional sentence pairs following the same pattern. We generate 130 such samples to be used to train the probes. Additionally we create 30 samples as our test set, which are manually checked by the authors to ensure that they follow the desired format.

In the initial setup, "so" cues "able" indicating causation, while "yet" cues "unable" indicating antithesis. To make the task more challenging, we introduce negative verbs in the satellite clause, which reverses the association between "able" and "unable" (Figure 3). For such verbs, "unable" typically fits causal contexts, whereas "able" is more appropriate for antithetical ones. This prevents the model from solely relying on cue words "so" or "yet" to distinguish between causation and antithesis. We suspect that the sentiment of the verb has an impact on the ability of the model to predict either "able" or "unable." For example, in Figure 1b, the verb sense is positive, whereas in Figure 3, the verb sense is negative.

Causation: She breaks rules, so she was *unable* to join the team.
Antithesis: She breaks rules, yet she was *able* to join the team.

Figure 3: An example illustrating how "unable" and "able" are used to differentiate causation and antithesis. Note that in Figure 1b, "able" appears in the causation example, whereas "unable" is used in the antithesis. This example demonstrates the opposite usage, highlighting how verb semantics can affect interpretation.

Moreover, verb semantics can shift under explicit negation. Therefore, for each case, we also generate examples with explicit negation. Introducing negation often reverses the use of 'able' and 'unable' in both causation and antithesis cases. For instance, the examples in Figure 3 are flipped in Figure 4 when 'not' is introduced in the satellite clause. We then examine model performance with these negated examples.

2.3 Models

We prompt two LLMs of comparable size and from the same architectural family in order to observe common patterns. Our primary model for investigation is **LLaMA** (Dubey et al., 2024), specifically `LLaMA-3-8B-Instruct`, an instruction-tuned version of LLaMA with 8 billion parameters, with 32 layers and 32 attention heads (AI@Meta, 2024). LlaMa uses grouped query attention with 8 key-value groups. We use **Mistral** as our second model (Jiang et al., 2023), specifically `Mistral-7B-Instruct-v0.2` (mistralai, 2024), an instruction-tuned version of Mistral. Similar to LLaMA, Mistral has 32 layers and 32 attention heads. However, Mistral employs sliding window attention (Beltagy et al., 2020; Zaheer et al., 2020).

Causation: She did not break rules, so she was *able* to join the team.
Antithesis: She did not break rules, yet she was *unable* to join the team.

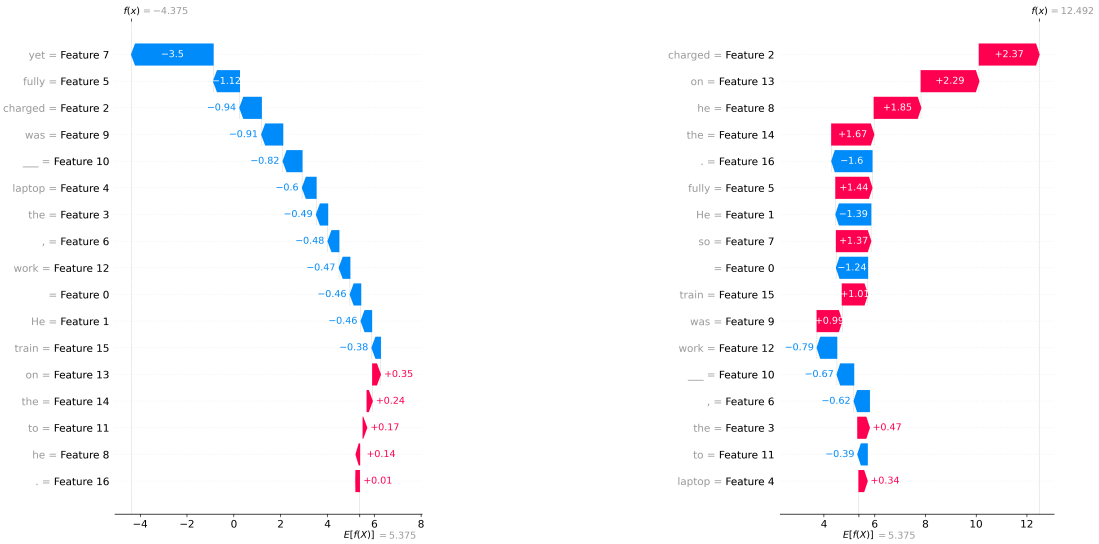
Figure 4: An example showing how "able" and "unable" switch roles when the satellite verb is negated. Examples correspond to those in Figure 3 and their negated forms.

We opt for instruction-tuned models given our use of modeling prompting in the experimental pipeline. Prior work has shown that instruction tuning does change LLM behavior, such as by altering the self-attention heads to attend more to verbs that appear commonly in prompts versus common verbs (Wu et al., 2024), making it unclear how our findings generalize to other foundation models.

2.4 Interpretability Methods

To address our research questions, we construct several techniques based in mechanistic interpretability. The first interpretability technique we operationalize is **Activation patching** (Geiger et al., 2021; Meng et al., 2022; Zhang and Nanda, 2024), which is a technique used to understand the causal effect of a specific internal representation on a model's predictions. In this approach, a portion of the model's internal representation is replaced with the representation generated from the model's processing of an alternative input. The causal role of the targeted unit is then assessed by analyzing the resulting changes in the output logits (i.e., the model's unnormalized scores for each possible output token). If substituting the representation shifts the logits or probits toward a different output, it indicates that the inspected unit plays a causal role in the model's prediction.

In our activation patching approach, we first run the model on a data instance x_{base} (either antithesis or causation) and measure the logit difference between the two answer choices, "able" vs "unable", which we denote as $LD(x_{base})$. Second, we run



(a) (b)

Figure 5: Normalized SHAP contributions for tokens across examples in LLaMA. Subfigures (a) and (b) illustrate example-level SHAP values for individual sentences.

the model on the corresponding counterexample x_{source}. The counterexample is created either by switching between "so"/"yet", or by negating the verb in the satellite. Therefore, the counterexample for antithesis will be causation, and vice versa. We then measure the logit difference between the two answer choices, denoted as $LD(x_{source})$. Next, we run the model on x_{base}, but for a target layer and token position, we replace the activation($h_{l,t}$) with the corresponding activation from the x_{source} run. We measure the logit difference as before, denoted by $LD(x_{base}; h^{base}_{l,t} \leftarrow h^{source}_{l,t})$. Following previous work (Wang et al., 2023; Li et al., 2025), we measure the *normalized logit difference* (NLD):

$$NLD = \frac{LD(x_{base}; h^{base}_{l,t}) - LD(x_{base})}{LD(x_{source}) - LD(x_{base})} \quad (1)$$

Second, we use **Logits Attribution** to assess the influence of internal layers in decision making. We measure the alignment of the layer representation with the *logit difference direction* (Wang et al., 2023; Janiak et al., 2024). Let the unembedding matrix be denoted as $W_{\text{unemb}} \in \mathbb{R}^{d \times |V|}$, where $|V|$ is the vocabulary size and d is the hidden dimension. For a hidden state representation $h_{L,t} \in \mathbb{R}^d$ at the L^{th} layer and time step t, the logits are given by $\ell = W^{\top}_{\text{unemb}} h_{L,t}$, which is a vector in $\mathbb{R}^{|V|}$. However, we are specifically interested in the logit difference between the correct answer (ca) and an incorrect answer (ia). This can be expressed as

$$\ell_{\text{ca}} - \ell_{\text{ia}} = \left(W_{\text{unemb}}[\texttt{ca}] - W_{\text{unemb}}[\texttt{ia}]\right)^{\top} h_{L,t}. \quad (2)$$

$W_{\text{unemb}}[\texttt{ca}]$ and $W_{\text{unemb}}[\texttt{ia}]$ denote the column vectors of the unembedding matrix for the correct and incorrect tokens. Thus, the logit difference reduces to the dot product between the hidden state representation $h_{L,t}$ and the *logit difference direction*. A higher dot product indicates a greater affinity of the layer toward the correct answer.

We also conduct three **probing** experiments (Shi et al., 2016; Alain and Bengio, 2017) on the internal representations of a frozen language model, where hidden states are extracted layer-wise, and a logistic regression classifier is trained as a probe:

- **"Able" vs. "unable"**: Layer-wise token representations are used to train a binary classifier that predicts whether the model's output corresponds to the answer *able* or *unable*.
- **Verb sentiment in the satellite** (sat-verb-sentiment): The probe is trained to classify the sentiment polarity (positive vs. negative) of the verb in the satellite clause (e.g., the verb in Figure 1b is positive, while that in Figure 3 is negative).
- **Overall sentiment of the satellite** (sat-clause-sentiment): The probe predicts the overall sentiment of the satellite clause, accounting for the effect of negation. For instance, the satellite in Figure 1b is positive, but negation reverses its polarity; conversely, in Figure 3 the clause is negative, but negation yields a positive sentiment.

Finally, to analyze information flow and quantify

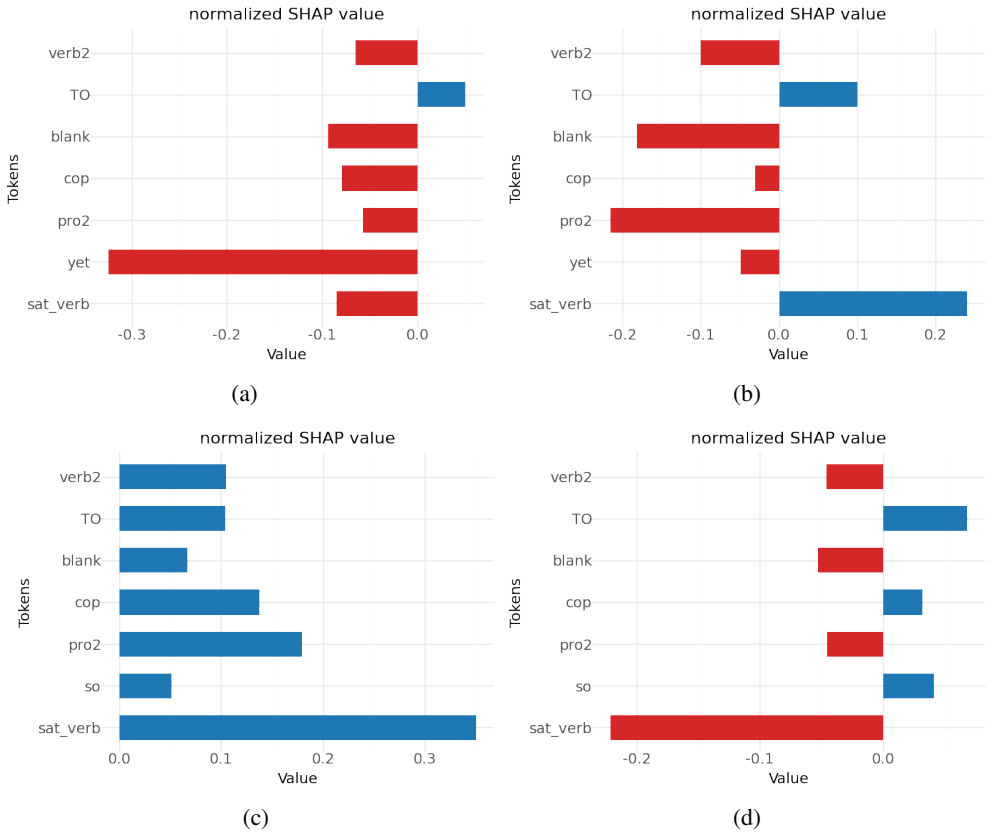


Figure 6: Normalized SHAP contributions for tokens across examples in LLaMA. The top row shows results for sentences containing the discourse marker "yet," and the bottom row shows results for sentences containing "so." Subfigures (a) and (c) present the average normalized SHAP values for positive verbs, while Subfigures (b) and (d) present the average normalized SHAP values for negative verbs.

edge contributions, we employ **attribution patching** (Nanda, 2023; Syed et al., 2024; Hanna et al., 2024). While *activation patching* provides a more precise intervention (Geiger et al., 2021; Meng et al., 2022; Zhang and Nanda, 2024), it is computationally expensive when identifying relevant subgraphs within the computation graph (Conmy et al., 2023; Syed et al., 2024). Attribution patching offers a more efficient alternative by approximating the effect of activation patching using a first-order Taylor expansion of the evaluation function around a base run, enabling scalable estimation of edge importance (Syed et al., 2024; Hanna et al., 2024).

$$L(x_{\text{base}} \mid \text{do}(E = e_{\text{source}})) \approx L(x_{\text{base}}) + \underbrace{(e_{\text{source}} - e_{\text{base}})^{\top} \frac{\partial}{\partial e_{\text{base}}} L(x_{\text{base}} \mid \text{do}(E = e_{\text{base}}))}_{\Delta_e L \text{ (attribution score)}} \quad (2)$$

Following (Syed et al., 2024), we compute absolute attribution scores and use them as measures of edge importance. We then select the *top-k* edges based on these scores.

2.5 Computational Resources

We run all experiments on a single NVIDIA Titan RTX GPU (24 GB VRAM). Edge attribution patching experiments are conducted on a single NVIDIA A100 GPU (80 GB VRAM) due to their higher memory requirements.

3 Results and Observations

The models perform well on our "able" versus "unable" task, with LLaMA achieving a zero-shot F1 score of 0.96, and Mistral achieving a zero-shot F1 score of 0.80. This high-performance enables effective interpretation of the models' encoding of discourse relations. We now present our experimental results addressing our research questions.

3.1 RQ1: Which token-level features influence the model's decisions?

To investigate which tokens have the greatest influence on the model's antithesis versus causation prediction, we first analyze the log-odds of the model's predictions using SHAP (Lundberg

and Lee, 2017). Figure 5a and Figure 5b present example-level SHAP values for individual sentences. Figure 6a and 6c show the average normalized average SHAP values for sentences containing "yet" and "so" paired with positive verbs, while Figure 6b and 6d show the corresponding values for sentences with negative verbs.

Notably, as illustrated in Figures 6b and 6c, the verb contributions are aligned: for positive verbs, "so" predominantly drives the decision towards "able," whereas for negative verbs, "yet" has a similar effect. Conversely, in Figure 6a and 6d, "yet" with positive verbs and "so" with negative verbs tend to lead to "unable" as the expected response. This supports our intuition regarding the influence of verb type at higher-level causal abstraction, as described in Section 2.2.

To quantify these differences, we perform a two-sample t-test across combinations of verb types and discourse markers—for example, comparing positive sentiment verbs with "so" and negative sentiment verbs with "yet" (Figure 6e and Figure 6c), as well as positive verbs with "yet" and negative sentiment verbs with "so" (Figure 6b and Figure 6f). The results indicate that the group means are significantly different, demonstrating that although the direction of the average contribution of "so" and "yet" aligns across verb types, their impact on model decisions is not identical. Depending on the verb type, specific tokens have a stronger influence in determining the output.[3]

Guided by these findings, we select our intervention locations at the "so"/"yet" token for the remaining research questions. Additionally, we choose the last token position, as higher-level semantic information is likely encoded there (Wang et al., 2023; Wiegreffe et al., 2025).

3.2 RQ2: Which layers play a pivotal role in decision making?

To understand which layers play a pivotal role in distinguishing antithesis and causation, we patch activations at the token position corresponding to "so"/"yet" and at the final token position.

Figure 7 illustrates the results of activation patching in the LLaMA model. Figure 16 in Appendix A depicts the same for the Mistral model. We observe that MLP layer activations are largely insensitive to the patching intervention(Figure 13). In contrast, attention and block outputs display meaningful patterns. Specifically, patching at the "so"/"yet" token position in early-to-mid attention layers transiently alters the output probits, whereas patching at the final token position affects probits in later mid layers. Notably, in both cases, interventions at higher layers fail to sustain this flipping effect. Similar trends are observed in the block outputs (Figure 7c).

This flipping of model output at early-mid layers for "so"/"yet" tokens suggests these layers encode local relational meanings (e.g., causal vs. contrastive) signaled by discourse markers. In contrast, higher layers appear to primarily propagate or refine the decision rather than reevaluate it, hence perturbations at this stage don't flip the output. Conversely, the sensitivity to patching at the last token position in the mid layers indicates that these layers integrate sentence-level meaning before producing logits. Once this integration is complete, later layers merely transmit the decision, rendering further interventions ineffective.

Interestingly, when patching is applied at the block output of the final token, the lower layers remain relatively stable, but the model's decision flips around layer 15. This aligns with our observations from RQ1: once mid-layer activations consolidate the model's decision, higher layers play a passive role, and swapping their activations with those from corrupted inputs cannot reverse the output. Similar observations are made when negating the satellite verb; see Figures 14 and 15 in Appendix A.

Building on these observations, we next identify which layers favor the correct answer. To do this, we analyze logit attribution by computing the dot product between each layer's output and the logit direction, where a positive result indicates that a layer's output pushes the model towards the correct answer. Figure 8a presents these attribution results for the attention outputs for positive verbs. Notably, layers such as 2, 3, 8, 10, and 12 tend to push the output in the negative direction—favoring "unable" irrespective of the right answer. In contrast, outputs from attention heads in layers 6, 7, 15, and 22 predominantly drive the prediction towards "able."

Figure 8b shows dot product values for 'able' across positive and negative sentiment verbs. While the overall magnitude of the dot product remains similar for both verb types, the distributions are sometimes bimodal (e.g., layers 3 and 5 for positive verbs, and layer 10 for negative verbs in Figure 8b). To understand the source of the bimodality we further plot for subclasses that answer 'able' for

[3] A detailed summary of t-test results for the verb position across all configurations is provided in Table 1.

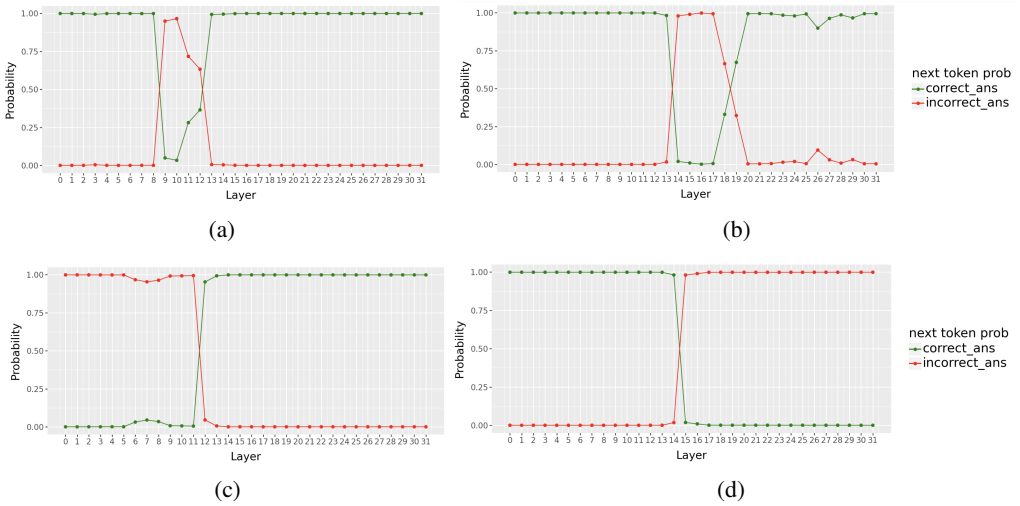


Figure 7: Change in probability from the correct answer ("able" or "unable") to the wrong answer under activation patching between causation and antithesis with a positive verb. The left column shows the change in probability when patched at the token location for "so" and "yet," while the right column shows the same change at the last token location. 7a and 7b show the effect of patching at the attention output, and 7c and 7d depict the effect of patching at the block output.

positive and negative verbs in Figures 8c and 8d. As discussed in subsection 2.2, for positive verbs, the answer 'able' can be elicited by the discourse marker "so," and by "yet" when the verb is negated with "not" (covering both causation and antithesis). The plots at layers 3 and 5 in Figure 8c clearly show two distinct kernels corresponding to each construction, accounting for the observed bimodality. A similar pattern for negative verbs is illustrated in Figure 8d.

3.3 RQ3: Which concepts aid model decision-making?

To identify which concepts aid model decision-making and evaluate the potential factors raised in Section 2.2, we next analyze the performance of logistic regression probes at each attention head and layer. Figure 9 shows probe accuracies for hidden representations at both the "so"/"yet" and end token positions. We train probes for every head-layer combination for the three classification tasks described in Section 2.4–"able" versus "unable," verb sentiment in the satellite, and overall sentiment of the satellite–allowing us to pinpoint which internal representations are most predictive and thus likely central to the model's reasoning process.

For the "able" vs. "unable" prediction at the "so"/"yet" token, the highest probe accuracies are found in heads from layers 8–16 (Figure 9a), with several heads in layers 6–7 also surpassing 0.5 accuracy. Notably, certain lower-level heads, such as head 12 and 20 in layer 3, achieve near-perfect accuracy. Similar patterns are observed in both *sat-verb-sentiment* and *sat-clause-sentiment* classification, where attention heads in the vicinity of layer 10 (± 5) consistently perform well. These outcomes highlight that the model is actively representing and encoding the high-level semantic cues such as verb sentiment and sentiment of the whole satellite clause. Additionally, the near-perfect accuracy of some lower-level heads indicates that early representations also contribute significantly to the decision process at the "so"/"yet" location. This is further supported by our activation patching analysis (Figure 7c), which demonstrates that perturbations at these positions can influence the model's output, confirming the functional importance of these encoded semantic concepts.

In contrast, for "able" vs. "unable" prediction at the final token, distinct performance patterns are less evident. Most heads after layer 11 demonstrate high accuracy, with layers 13 and 14 containing several heads approaching perfect classification. This aligns with our previous observation that activation patching at these higher layers' attention can briefly alter the output (Figure 9e), suggesting strong decision signals. Interestingly, some heads in upper layers exhibit lower activity for classification, implying a potential role in supporting or refining the decision process.

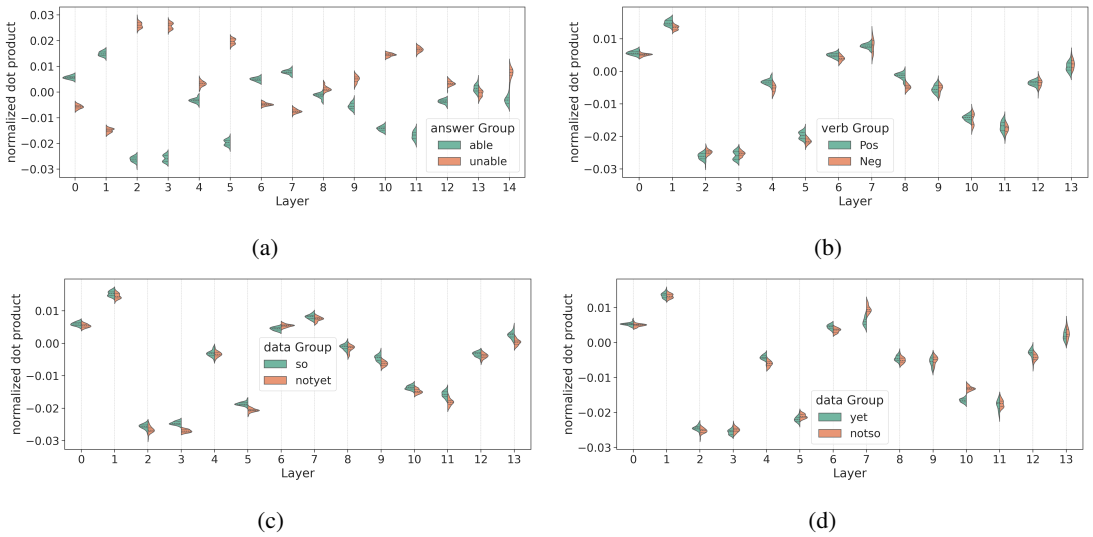


Figure 8: Distribution of dot products between layer outputs and logit difference direction. (a) layerwise dot product distributions for 'able' vs. 'unable' answers for positive verbs. (b): comparisons across layers 0–13 for different verb types for answer type 'able.' (c): For positive verb type comparisons across layers 0–13 for different combinations with correct answer 'able.' This explains the bimodality presented in (b). (d): As c for negative verbs.

3.4 RQ4: How does information propagate

To understand how information propagates through the model layers, we employ edge attribution patching (EAP) as described in subsection 2.4. To address the issue of vanishing gradients (Syed et al., 2024; Hanna et al., 2024), we also apply integrated gradients (IG) (Sundararajan et al., 2017; Hanna et al., 2024) in conjunction with attribution patching (EAP-IG). Experiments are conducted using both 0-shot and few-shot settings. For each setting, we report the top 1,000 edges identified. Comparing the intersection of edges from both experiments, we find that 51% of the edges are shared, suggesting a stable core of influential connections that consistently support information propagation, regardless of the shot setting.

When analyzing negative verbs, the overlap across shot settings drops to 33.9%, indicating greater instability in the set of important edges. This raises the question of whether the model relies on different pathways to process verbs with negative sentiment compared to verbs with positive sentiment, suggesting that EAP-IG may be more sensitive to shot configuration or verb type in these cases.

Thus, to ascertain whether the model relies on different pathways to process negative verbs, we keep the shot configuration fixed and compare edge overlap across verb types. The relatively high overlaps (60.8% for 0-shot; 79.4% for 2-shot) indicate that information propagation patterns are more consistent across verb types within the same prompt setup, especially when more context (2-shot) is provided.

4 Related Work

Many techniques have been proposed to uncover a model's inner workings by reverse engineering its computational processes (Geiger et al., 2021, 2024; Meng et al., 2022; Wu et al., 2023; Conmy et al., 2023; Nanda, 2023; Syed et al., 2024; Hanna et al., 2024; Zhang and Nanda, 2024). These mechanistic interpretability methods have been applied to reveal model behavior in tasks such as indirect object identification, the greater-than task, question answering, and more (Wang et al., 2023; Hanna et al., 2023; Wiegreffe et al., 2025). However, this line of work, while providing insights into model internals, often overlooks the rich linguistic structures underlying text data.

Recently, researchers have begun to explore model understanding with construction grammar (Scivetti et al., 2025; Scivetti and Schneider, 2025; Boguraev et al., 2025; Rozner et al., 2025), integrating deeper linguistic theory into the analysis of LLMs. In this paper, we advance this line of inquiry by analyzing models from a discourse perspective. Specifically, we examine model behavior through the lens of Rhetorical Structure Theory (RST; Mann and Thompson, 1987; Mann et al.,

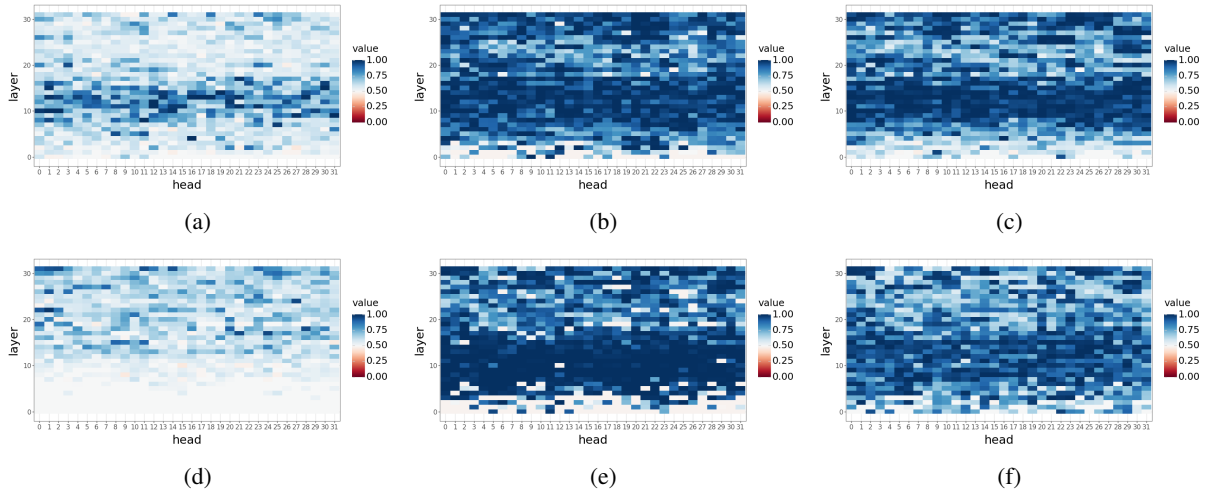


Figure 9: Accuracy of logistic regression probes across attention heads of LLaMA. The top row shows probe accuracy at the token position for "so"/"yet", while the bottom row shows accuracy at the position of the last token in the input. Subfigures (a) and (d) show classification accuracy for *"able"* vs. *"unable"*; (b) and (e) show accuracy for classifying the sentiment polarity of the satellite verb; and (c) and (f) show accuracy for predicting the sentiment of the satellite clause.

1989). We opt to use RST (as opposed to other discourse modeling frameworks such as PDTB (Webber et al., 2019)) due to its structured yet global approach. Additionally, its nucleus-satellite construction eases delineation discourse scope.

Most related to our work, Miao and Kan (2025) recently investigate factors promoting discourse understanding in Transformers, with a focus on circuit discovery. Similarly to our approach, they use a next-token prediction task with constrained options (e.g., Arg2 or Arg‘2) to study how circuits encode discourse relations through "discursive circuits." In contrast, our work takes a broader perspective by applying multiple interpretability techniques and focusing on two key RST relations: causation and antithesis.

RST has been leveraged to enhance the explainability of LLMs by fusing RST into neural networks, for cross-lingual natural language inference (Yuan et al., 2025) plus entailment reasoning for science question answering (Zhang et al., 2024).

5 Conclusion

In this paper, we investigate Transformer encoding of discourse relations, which are critical to characterizing the structure of (and thus to understanding) a document.

We frame this task as a next-token prediction task distinguishing causation and antithesis, and apply a range of causal interventions and probing techniques. Our results indicate that early to mid layers are crucial for decision making at discourse markers such as 'so' or 'yet,' while higher layers at the final token position are responsible for making and maintaining the model's ultimate prediction. At the circuit level, we observe a stable set of connections that reliably contribute to computation, though additional computational edges may be contextually activated or deactivated depending on the prompt and verb type. Furthermore, we find that certain layers exhibit inherent biases toward specific outcomes. Our results provide insight into how discourse modeling is promoted in language models, giving way to broader model safety and control. Further, the statistically significant and substantive native of our findings suggests that our methodology is extensible to additional discourse relations and model architectures.

6 Limitations

Like prior work (Miao and Kan, 2025), we constrain our task to a next-token prediction task with one of two expected outputs and focus on two RST relations. We opt to study model behavior through the lens of RST. More specifically our analysis is on distinguishing causation and antithesis, formulated as a next-token prediction task. Future work may adapt our approach to examine additional RST relations.

We focus our analyses on Transformer models due to their prevalence in contemporary NLP literature, and in order to promote generalizability we

examine both a LLaMA and Mistral model.

This study focuses exclusively on discourse relations in English. As discourse structure and linguistic cues can vary substantially across languages, generalization of our results across languages is unknown. Furthermore, the observed results may depend on the specific models and discourse relations examined in this work.

References

Rilwan Adewoyin, Ritabrata Dutta, and Yulan He. 2022. RSTGen: Imbuing fine-grained interpretable control into long-FormText generators. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1822–1835, Seattle, United States. Association for Computational Linguistics.

AI@Meta. 2024. Llama 3 model card.

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Sasha Boguraev, Christopher Potts, and Kyle Mahowald. 2025. Causal interventions reveal shared structure across English filler–gap constructions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25032–25053, Suzhou, China. Association for Computational Linguistics.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. 2024. Finding alignments between interpretable causal variables and distributed neural representations. 236:160–187.

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 *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *ICML 2024 Workshop on Mechanistic Interpretability*.

Vrindavan Harrison, Lena Reed, Shereen Oraby, and Marilyn Walker. 2019. Maximizing stylistic control and semantic accuracy in NLG: Personality variation and discourse contrast. In *Proceedings of the 1st Workshop on Discourse Structure in Neural NLG*, pages 1–12, Tokyo, Japan. Association for Computational Linguistics.

Zehra Melce Hüsünbeyi, Didar Akar, and Arzucan Özgür. 2022. Identifying hate speech using neural networks and discourse analysis techniques. In *Proceedings of the First Workshop on Language Technology and Resources for a Fair, Inclusive, and Safe Society within the 13th Language Resources and Evaluation Conference*, pages 32–41, Marseille, France. European Language Resources Association.

Jett Janiak, Can Rager, James Dao, and Yeu-Tong Lau. 2024. An adversarial example for direct logit attribution: Memory management in GELU-4L. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 232–237, Miami, Florida, US. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Zae Myung Kim, Anand Ramachandran, Farideh Tavazoee, Joo-Kyung Kim, Oleg Rokhlenko, and Dongyeop Kang. 2026. Align to structure: Aligning large language models with structural information. 40(44):37519–37528.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. 2025. (how) do language models track state? In *Forty-second International Conference on Machine Learning*.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

William Mann, Christian Matthiessen, and Sanda Thompson. 1989. Rhetorical structure theory and text analysis. *Discourse Description: Diverse Linguistic Analyses of a Fund Raising Text*, page 66.

William C. Mann and Sandra A. Thompson. 1987. Rhetorical structure theory: Description and construction of text structures.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Yisong Miao and Min-Yen Kan. 2025. Discursive circuits: How do language models understand discourse relations? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32558–32577, Suzhou, China. Association for Computational Linguistics.

mistralai. 2024. Mistral-7b-instruct-v0.2 model card.

Nishanth Sridhar Nakshatri, Nikhil Mehta, Siyi Liu, Sihao Chen, Daniel Hopkins, Dan Roth, and Dan Goldwasser. 2025. Talking point based ideological discourse analysis in news events. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 575–594, Vienna, Austria. Association for Computational Linguistics.

Neel Nanda. 2023. Attribution patching: Activation patching at industrial scale. https://www.neelnanda.io/mechanistic-interpretability/attribution-patching. Accessed: 2025-10-02.

Joshua Rozner, Leonie Weissweiler, Kyle Mahowald, and Cory Shain. 2025. Constructions are revealed in word distributions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2116–2138, Suzhou, China. Association for Computational Linguistics.

Wesley Scivetti, Tatsuya Aoyama, Ethan Wilcox, and Nathan Schneider. 2025. Unpacking *Let Alone*: Human-scale models generalize to a rare construction in form but not meaning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27491–27502, Suzhou, China. Association for Computational Linguistics.

Wesley Scivetti and Nathan Schneider. 2025. Construction identification and disambiguation using BERT: A case study of NPN. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 365–376, Vienna, Austria. Association for Computational Linguistics.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Aaquib Syed, Can Rager, and Arthur Conmy. 2024. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 407–416, Miami, Florida, US. Association for Computational Linguistics.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. 2019. The penn discourse treebank 3.0 annotation manual. *Philadelphia, University of Pennsylvania*, 35:108.

Sarah Wiegreffe, Oyvind Tafjord, Yonatan Belinkov, Hannaneh Hajishirzi, and Ashish Sabharwal. 2025. Answer, assemble, ace: Understanding how lms answer multiple choice questions. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Xuansheng Wu, Wenlin Yao, Jianshu Chen, Xiaoman Pan, Xiaoyang Wang, Ninghao Liu, and Dong Yu. 2024. From language modeling to instruction following: Understanding the behavior shift in LLMs after instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2341–2369, Mexico City, Mexico. Association for Computational Linguistics.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah D. Goodman. 2023. Interpretability at scale: identifying causal mechanisms in alpaca. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Mengying Yuan, WenHao Wang, Zixuan Wang, Yujie Huang, Kangli Wei, Fei Li, Chong Teng, and Donghong Ji. 2025. Cross-document cross-lingual NLI via RST-enhanced graph fusion and interpretability prediction. In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, pages 11–33, Suzhuo, China. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: transformers for

longer sequences. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

Longyin Zhang, Bowei Zou, and Ai Ti Aw. 2024. Empowering tree-structured entailment reasoning: Rhetorical perception and LLM-driven interpretability. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5783–5793, Torino, Italia. ELRA and ICCL.

A Appendix

System Prompt: You are an AI language model trained to choose between the words "able" and "unable" to complete a partially written sentence. Your decision should be based on the meaning of the sentence and its implied relationship, such as causation or contrast.
Carefully review the provided examples. Then, for each new sentence, output only the correct word: "able" or "unable."
Do not provide explanations. Do not output anything else.

User Prompt: 1. Sentence: John got the train this morning, so he was ___ to reach on time.
Answer: able
2. Sentence: The team practiced all week, so they were ___ to perform well.
Answer: able
3. Sentence: The team practiced all week, yet they were ___ to perform well.
Answer: unable
4. Sentence: Lena studied hard, yet she was ___ to pass the exam.
Answer: unable

Now complete the following:
5. Sentence: He slept well last night, yet he was ___ to run this morning.
Answer:

Figure 10: Example with system prompt and few-shot demonstrations.

Number of demonstrations per prompt. For prompting the model to predict the last token we replace the actual 'able' or 'unable' token from the query sentence and present the query sentence along with some example sentences. An example with the system prompt is demonstrated in Figure 10. We use two demonstrations per sentence type—'so,' 'yet,' 'NOT-so,' and 'NOT-yet.' Following (Wiegreffe et al., 2025), we patch only those examples for which the model predicts accurately. To understand the effect of the number of demonstrations in the prompt, we experimented with up to four examples per class. Figure 11 shows the comparative recall and precision for next-token predictions of 'able' vs. 'unable' with the LLaMA model. Interestingly, the model performs better without any demonstrations: for instances without negation, recall is consistently high for 'able,' and precision is high for 'unable.' This pattern reverses when explicit negation ('not') is introduced in the satellite verb.

We also observe that the order of demonstrations in the prompt impacts performance. Consistency between the query and the provided demonstrations yields perfect recall and precision. For example, queries without 'NOT' benefit from demonstrations without 'NOT,' and queries with 'NOT' benefit from demonstrations with 'NOT.' In this setup, we use two shots per class. While the organization, order, and number of demonstrations could be systematically studied as a separate problem, we leave this for future work.

	pos-so	pos-yet	negated-pos-so	negated-pos-yet	neg-so	neg-yet	negated-neg-so	negated-neg-yet
pos-so	-	diff	diff	diff	diff	diff	diff	diff
pos-yet	diff	-	diff	not diff	diff	not diff	diff	diff
negated-pos-so	diff	diff	-	diff	diff	diff	not diff	diff
negated-pos-yet	diff	not diff	diff	-	diff	not diff	diff	diff
neg-so	diff	diff	diff	diff	-	diff	diff	not diff
neg-yet	diff	not diff	diff	not diff	diff	-	diff	diff
negated-neg-so	diff	diff	not diff	diff	diff	diff	-	diff
negated-neg-yet	diff	diff	diff	diff	not diff	diff	diff	-

Table 1: Pairwise t-test of normalized SHAP values across constructions at verb position. ‘pos’ is for verb with positive sentiment. ‘neg’ is for verb with negative sentiment. ‘pos-so’ refers to sentences with positive verb and so as markers. ‘negated-pos-so’ refers to example with explicit negation of positive verb and ‘so’ as discourse marker.

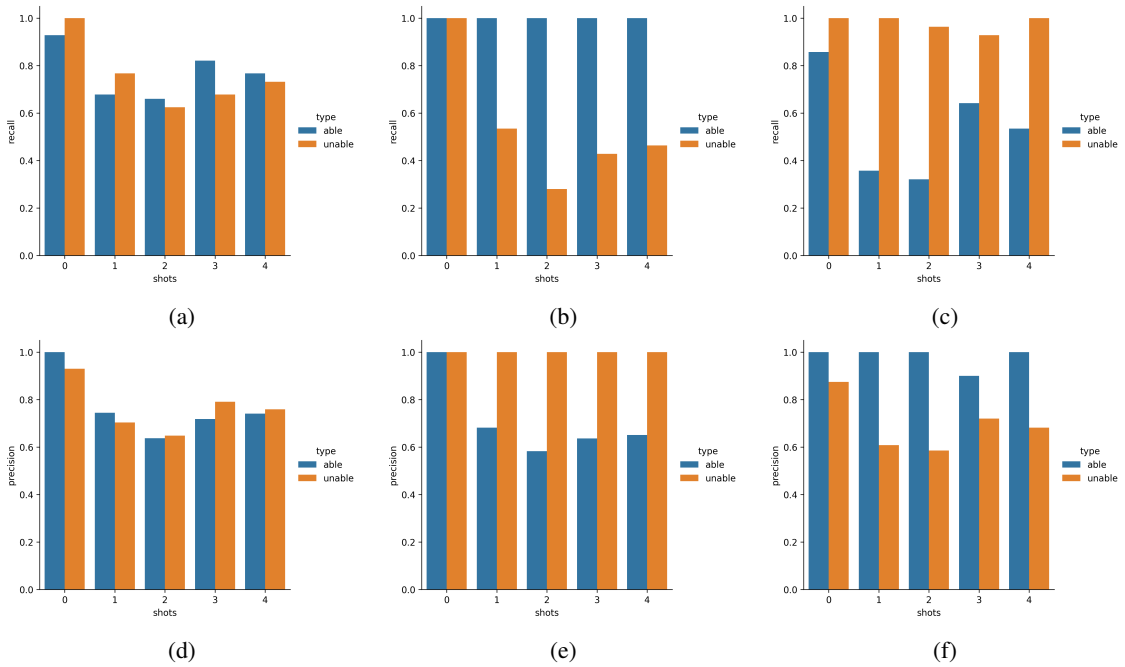


Figure 11: LLaMA performance on examples with positive verbs. The top row shows recall, and the bottom row shows precision. (a) and (d): overall performance; (b) and (e): without explicit negation; (c) and (f): with negation.

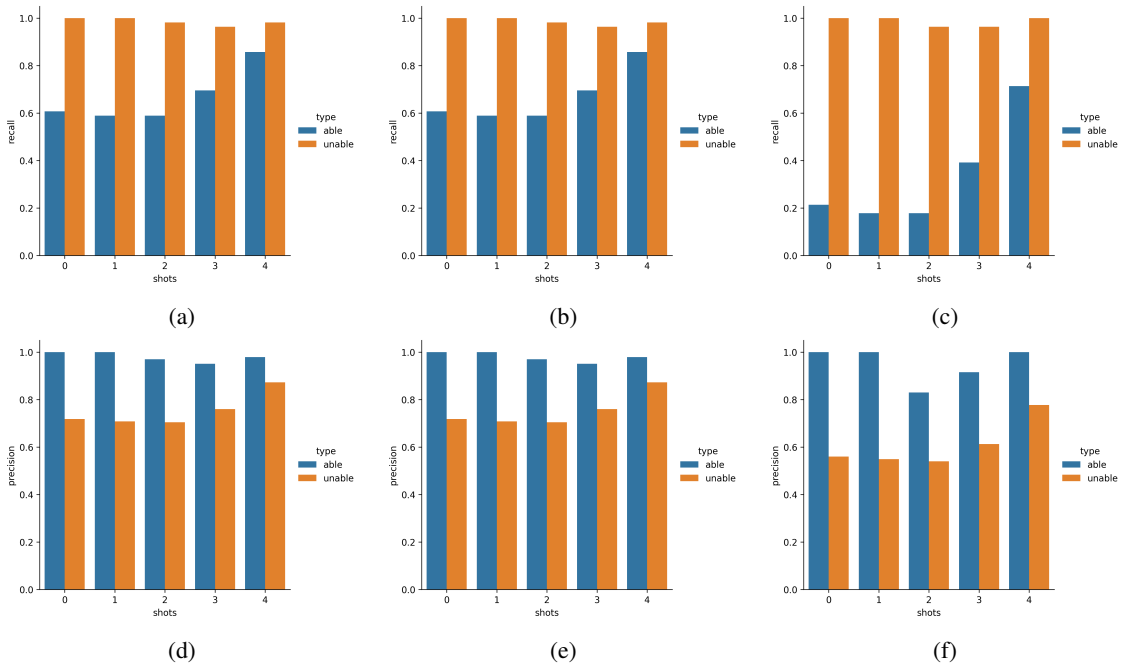


Figure 12: Mistral performance on examples with positive verbs. The top row shows recall, and the bottom row shows precision. (a) and (d): overall performance; (b) and (e): without explicit negation; (c) and (f): with negation.

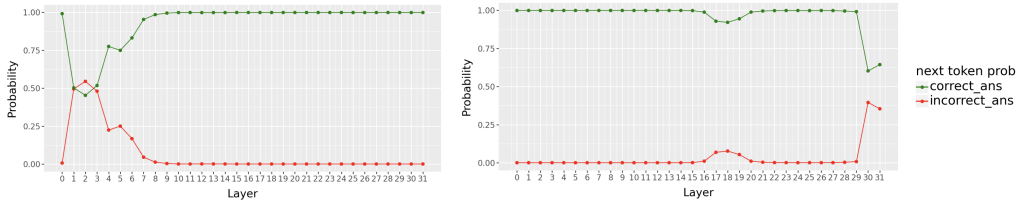


Figure 13: Change in probability from the correct answer ("able" or "unable") to the wrong answer under activation patching between causation and antithesis with a positive verb. The left column shows the change in probability when patched at the token location for "so" and "yet," while the right column shows the same change at the last token location. (a) and (b) illustrate the effect of patching at the MLP activation.

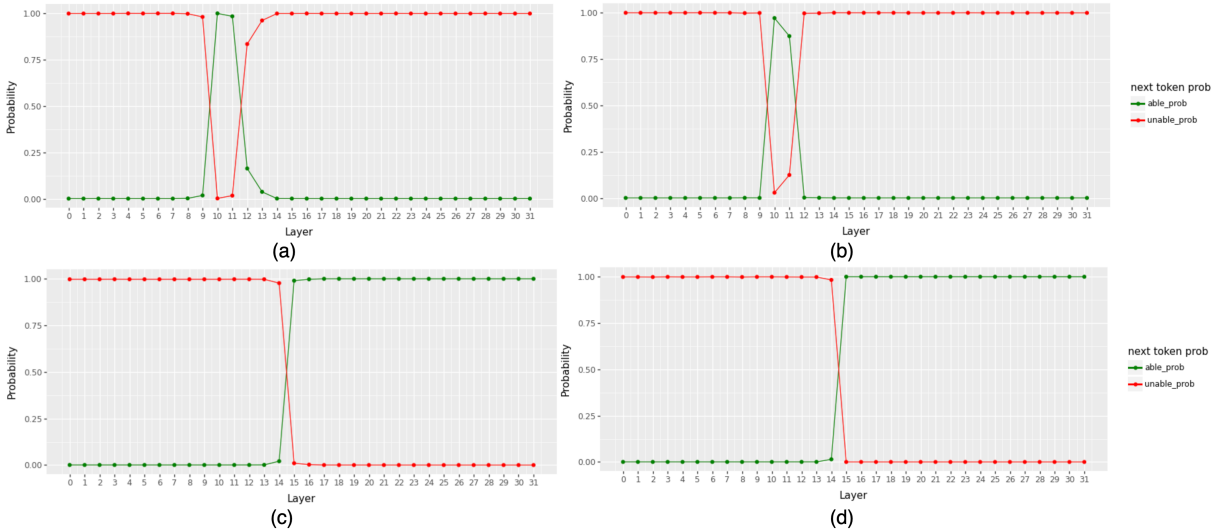


Figure 14: Activation patching in LLaMA with sentence with explicit ‘NOT’ and ’unable’ is the correct answer. Top rows depicts when patched at “so”/“yet” location. Bottom row shows when patched at last token. (a) and (c)patching from source combination not-yet(able) to base combination yet(unable) (b) and (d) patching from source combination so(able) to base combination not-so(unable)

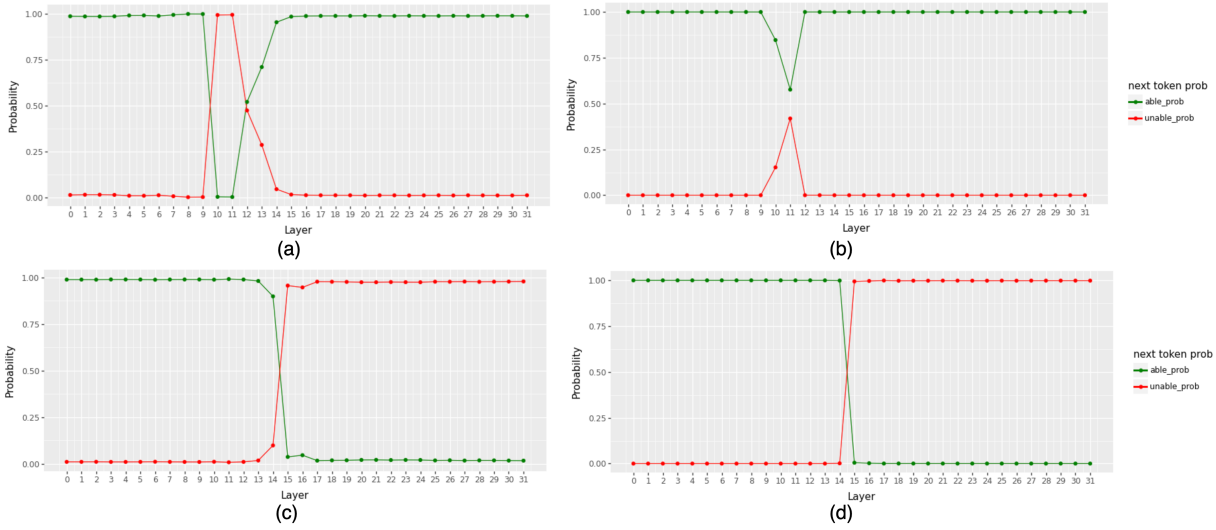


Figure 15: Activation patching in LLaMA with sentence with explicit ‘NOT’ and ’able’ is the correct answer. Top rows depicts when patched at “so”/“yet” location. Bottom row shows when patched at last token. (a) and (c)patching from source combination not-yet(able) to base combination yet(unable) (b) and (d) patching from source combination so(able) to base combination not-so(unable)

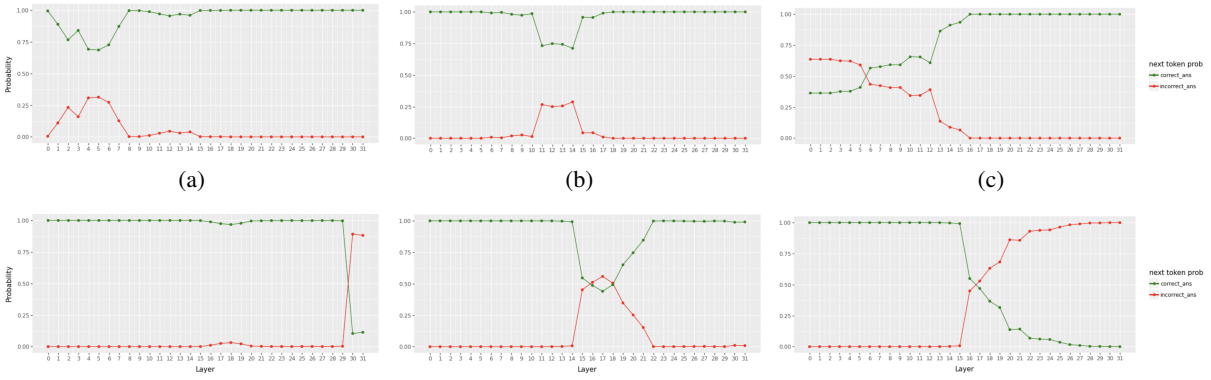


Figure 16: Change in probability in Mistral model from the correct answer (able or unable) to the wrong answer (unable or able) under activation patching between causation and antithesis with a positive verb. The top row shows the change in probability when patched at the token location for "so" and "yet", while the bottom row shows the same change at the last token location. (a) and (d) illustrate the effect of patching at the MLP activation. (b) and (e) show the effect of patching at the attention output, and (c) and (f) depict the effect of patching at the block output.

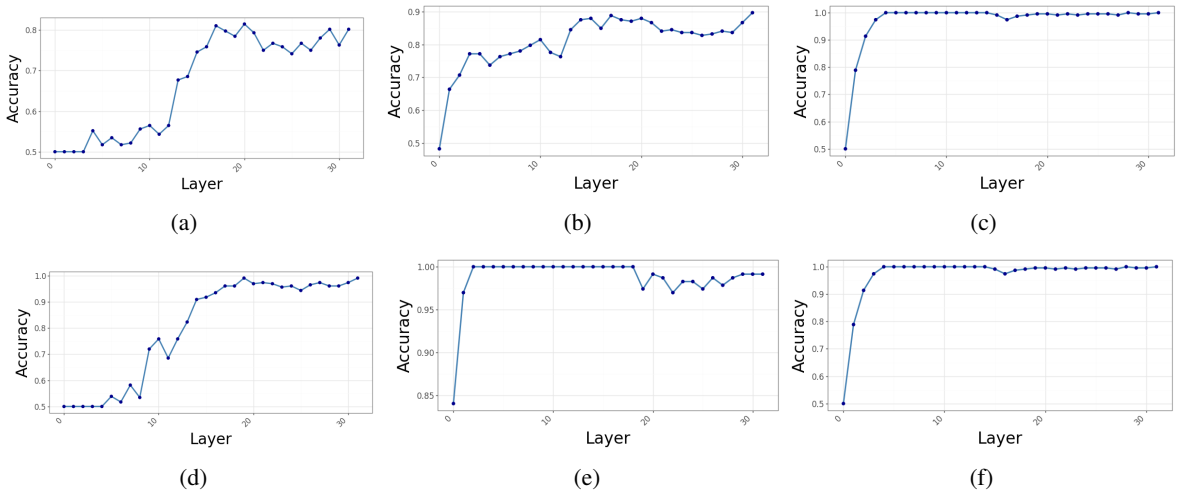
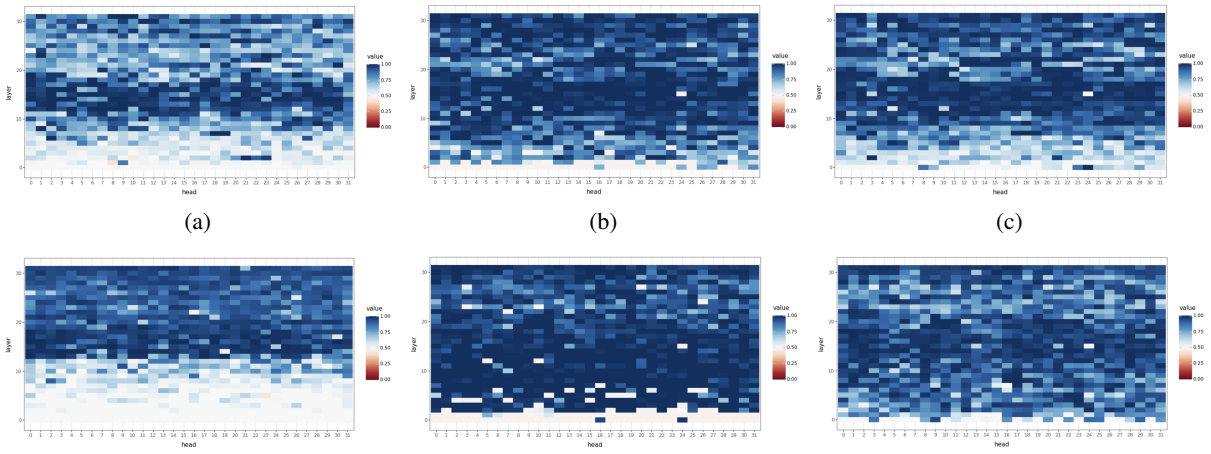


Figure 17: Accuracy of logistic regression probes across MLP layers of LLaMA. Top row shows probe accuracy at the last token position for 0-shot prompt. Bottom row shows probe accuracy at the last token position for 2-shots per class. (a) and (d) show classification accuracy for "able" vs. "unable." (b) and (e) show accuracy for classifying the sentiment polarity of the satellite verb; and (c) and (f) show accuracy for predicting the sentiment of the satellite clause



(a) (b) (c)

(d) (e) (f)

Figure 18: Accuracy of logistic regression probes across attention heads of Mistral. The top row shows probe accuracy at the token position for "so"/"yet", while the bottom row shows accuracy at the position of the last token in the input. Subfigures (a) and (d) show classification accuracy for *"able"*vs. *"unable"*; (b) and (e) show accuracy for classifying the sentiment polarity of the satellite verb; and (c) and (f) show accuracy for predicting the sentiment of the satellite clause.