Reasoning Circuits in Language Models: A Mechanistic Interpretation of Syllogistic Inference

Geonhee Kim[1], **Marco Valentino**[2,3], **André Freitas**[1,2,4]
[1]Department of Computer Science, University of Manchester, UK
[2]Idiap Research Institute, Switzerland
[3]School of Computer Science, University of Sheffield, UK
[4]National Biomarker Centre, CRUK-MI, University of Manchester, UK
kimgeonhee317@gmail.com, ac4mv@sheffield.ac.uk, andre.freitas@idiap.ch

Abstract

Recent studies on reasoning in language models (LMs) have sparked a debate on whether they can learn systematic inferential principles or merely exploit superficial patterns in the training data. To understand and uncover the mechanisms adopted for formal reasoning in LMs, this paper presents a mechanistic interpretation of syllogistic inference. Specifically, we present a methodology for circuit discovery aimed at interpreting content-independent and formal reasoning mechanisms. Through two distinct intervention methods, we uncover a sufficient and necessary circuit involving middle-term suppression that elucidates how LMs transfer information to derive valid conclusions from premises. Furthermore, we investigate how belief biases manifest in syllogistic inference, finding evidence of partial contamination from additional attention heads responsible for encoding commonsense and contextualized knowledge. Finally, we explore the generalization of the discovered mechanisms across various syllogistic schemes, model sizes and architectures. The identified circuit is sufficient and necessary for syllogistic schemes on which the models achieve high accuracy ($\geq$ 60%), with compatible activation patterns across models of different families. Overall, our findings suggest that LMs learn transferable content-independent reasoning mechanisms, but that, at the same time, such mechanisms do not involve generalizable and abstract logical primitives, being susceptible to contamination by the same world knowledge acquired during pre-training.

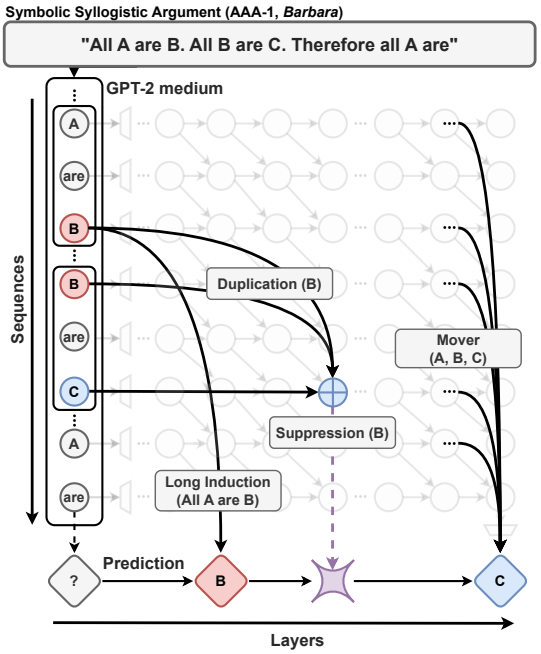

Figure 1: Conceptual representation of the circuit for processing symbolic syllogisms: (Long Induction) Early layers exhibit biases towards the wrong conclusion due to long-range repetition of the first premise "All A are B". (Duplication) Induction heads aggregate information about duplicated middle terms. (Suppression) The model inhibits middle-term information (i.e., 'B'), suppressing the long induction mechanism. (Mover) Token-specific information is propagated to the last token position. The process culminates in the prediction shift from 'B' to the correct token, 'C.'

1 Introduction

Language models (LMs) have led to remarkable results across various natural language processing tasks (Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2024; Jason et al., 2022; Bubeck et al., 2023). This success has catalyzed research interest in systematically exploring the reasoning capabilities emerging during pre-training (Clark et al., 2020). Recent findings suggest that logical and formal reasoning abilities may emerge in large-scale models (et al., 2022; Kojima et al., 2022; Wei et al., 2022) or through transfer learning on specialized datasets (Betz et al., 2021). However, ongoing debates question whether these models apply systematic inference rules or reuse superficial patterns learned during pre-training (Talmor et al., 2020; Kassner et al., 2020; Wu et al., 2024).

Findings of the Association for Computational Linguistics: ACL 2025, pages 10074–10095
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

This controversy underscores the need for a deeper understanding of the low-level logical inference mechanisms in LMs (Rozanova et al., 2024, 2023, 2022; Yanaka et al., 2020).

To improve our understanding of the internal mechanisms, this paper focuses on mechanistic interpretability (Olah et al., 2020; Nanda et al., 2023), aiming to discover the core circuit responsible for syllogistic reasoning. In particular, we focus on categorical syllogisms with universal affirmative quantifiers (i.e., AAA-1, *Barbara*) motivated by two key factors. First, as observed in natural logic studies (MacCartney and Manning, 2007), this form of syllogistic reasoning is prevalent in everyday language. Therefore, it is likely that LMs are exposed to such reasoning schema during pre-training. Second, AAA-1 is a form of unconditionally valid syllogism independent of the premises' truth condition (Holyoak and Morrison, 2005). This characteristic offers a deterministic and scalable task design, other than allowing us to investigate the disentanglement between reasoning and knowledge representation (Bertolazzi et al., 2024; Wysocka et al., 2025; Lampinen et al., 2024; Valentino et al., 2025).

Through mechanistic interpretability techniques such as Activation Patching (Meng et al., 2022) and embedding space analysis (i.e., Logit Lens) (Nostalgebraist, 2020; Geva et al., 2022; Dar et al., 2023), we investigate the following main research questions: *RQ1: How is the content-independent syllogistic reasoning mechanism internalized in LMs during pre-training?*; *RQ2: Are content-independent mechanisms disentangled from specific world knowledge and belief biases?*; *RQ3: Does the core reasoning mechanism generalize across syllogistic schemes, different model sizes, and architectures?*

To answer these questions, we present a methodology that consists of 3 main stages. First, we define a syllogistic completion task designed to assess the model's ability to predict valid conclusions from premises and facilitate the construction of test sets for circuit analysis. Second, we implement a circuit discovery pipeline on the syllogistic schema instantiated only with symbolic variables (Table 1, Symbolic) to identify the core sub-components responsible for content-independent reasoning. We conduct this analysis under two intervention methods: *middle-term corruption* and *all-term corruption*, aiming to identify latent transitive reasoning mechanisms and term-related information flow. Third, we investigate the generalization of the identified circuit on concrete schemes instantiated with commonsense knowledge to identify potential belief biases and explore how the internal behavior varies with different schemes and model sizes.

We present the following overall conclusions:

1. The circuit analysis reveals that LMs develop specific inference mechanisms during pre-training, finding evidence supporting a three-stage mechanism for syllogistic reasoning: (1) naive recitation of the first premise; (2) suppression of duplicated middle-term information; and (3) mediation towards the correct output through the interplay of mover attention heads (see Figure 1).

2. Further experiments on circuit transferability demonstrate that the identified mechanism is still necessary for reasoning on syllogistic schemes instantiated with commonsense knowledge. However, a deeper analysis suggests that specific belief biases acquired during pre-training might contaminate the content-independent circuit mechanism with additional attention heads responsible for encoding contextualized world knowledge.

3. We found that the identified circuit is sufficient and necessary for all the unconditionally valid syllogistic schemes in which the model achieves high downstream accuracy ($\geq$ 60%) (see Appendix A for the list of schemes). This result suggests that LMs learn reasoning mechanisms that are transferable across different schemes.

4. The intervention results on models with different architectures and sizes (i.e., GPT-2 (Radford et al., 2019), Pythia (Biderman et al., 2023), Llama (et al., 2024), Qwen (Yang et al., 2024)) show similar suppression mechanism patterns and information flow. However, we found evidence that the contribution of attention heads becomes more complex with model sizes, further supporting the hypothesis of increasing contamination from external world knowledge.

Our study is conducted using TransformerLens (Nanda and Bloom, 2022) on an Nvidia A100 GPU with 80GB of memory. The dataset and code to reproduce our experiments are available online to encourage future work in the field[1].

2 Methodology

Our main research objective is to discover and interpret the core mechanisms adopted by auto-

[1] https://github.com/neuro-symbolic-ai/Mechanistic-Interpretation-Syllogism

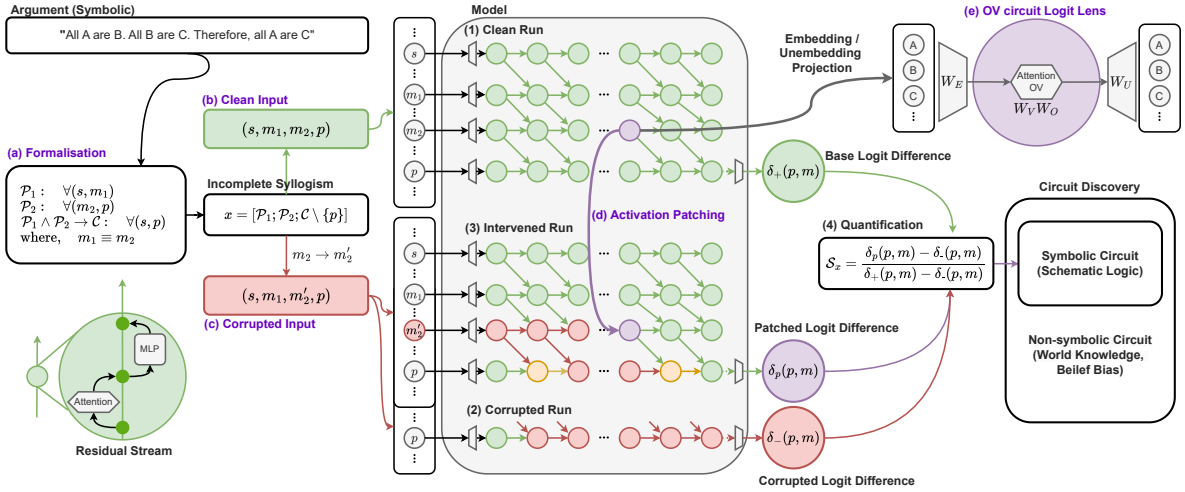

Figure 2: The conceptual pipeline of the circuit discovery methodology. Figures (a)–(e) illustrate the stages in processing a syllogistic argument: formalization (a), construction of clean (b) and corrupted inputs (c), activation patching (d) and embedding space analysis (i.e., Logit Lens) (e). The uncovered symbolic circuit is evaluated to determine whether the core reasoning schema operates independently from world knowledge or belief bias.

Premises ($\mathcal{P}_1, \mathcal{P}_2$)	**Conclusion** ($\mathcal{C}$)
Symbolic	
All A are B. All B are C.	Therefore, all A are **C**.
Belief-consistent (True premises)	
All men are humans. All humans are mortal.	Therefore, all men are **mortal**.
Belief-inconsistent (False premises)	
All pilots are people. All people are blond.	Therefore, all pilots are **blond**.

Table 1: Examples of a syllogistic schema (i.e., AAA-1, Barbara). The logical validity of a conclusion is only a function of the reasoning schema, being independent of the specific variables or truth condition of the premises.

regressive language models (LMs) when performing content-independent syllogistic reasoning. To this end, we present a methodology that consists of 3 main stages. First, we define a syllogistic completion task that can be instrumental for our analysis. Second, we leverage the syllogistic completion task to implement a circuit discovery pipeline on a syllogistic schema instantiated only with symbolic variables (Table 1, Symbolic). Third, we investigate the generalization of the identified circuit on concrete schemes instantiated with commonsense knowledge and explore how the internal behavior varies with different schemes and models.

2.1 Syllogism Completion Task

We design a syllogism completion task for assessing the reasoning abilities of LMs, building upon established approaches in the literature (Betz et al., 2021; Wu et al., 2023). In particular, we focus on categorical syllogisms with universal affirmative quantifiers (i.e., AAA-1, *Barbara*) because of their frequency in natural language and their content-independent reasoning property. The syllogistic argument is typically composed of two premises ($\mathcal{P}_1, \mathcal{P}_2$) and a conclusion ($\mathcal{C}$), with s and p denoting the subject and predicate terms in the conclusion (e.g. *men* and *mortal*), and m_1 and m_2 (e.g., *humans*) denoting the middle terms in the two premises, with $m_1 \equiv m_2$ in the case of the AAA-1 syllogism.

To evaluate syllogistic reasoning in LMs, we formalize the completion task as language modeling, removing the final token p from the conclusion (e.g., *All men are humans. All humans are mortal. Therefore, all men are*), and comparing the probability assigned by the LM to p (e.g., *mortal*) and the middle term m (e.g., *humans*). In general, an LM is successful in the completion of a task if the following condition applies:

$$P(p \mid [\mathcal{P}_1; \mathcal{P}_2; \mathcal{C} \backslash \{p\}]) > P(m \mid [\mathcal{P}_1; \mathcal{P}_2; \mathcal{C} \backslash \{p\}])$$

In our experiments, we measure the logit difference δ between the tokens p and m, defined as $\delta(p, m) = \text{logit}(p) - \text{logit}(m)$, which approximates the log ratio of the conditional probabilities:

$$\delta(p, m) \approx \log \left(\frac{P(p \mid [\mathcal{P}_1; \mathcal{P}_2; \mathcal{C} \setminus \{p\}])}{P(m \mid [\mathcal{P}_1; \mathcal{P}_2; \mathcal{C} \setminus \{p\}])} \right)$$

Dataset Construction. We conduct experiments using two distinct datasets, symbolic and non-symbolic, to derive comparative implications for model reasoning capabilities. The symbolic dataset is constructed by randomly sampling triples (e.g., A, B, and C) from the set of 26 uppercase letters of the English alphabet. On the other side, the non-symbolic dataset is constructed by replacing the letters with words while preserving the syllogistic schema. Here, we generate two different non-symbolic sets: a belief-consistent set containing true premises and a belief-inconsistent set containing false premises to address RQ2 (see Table 1). To guarantee the truth condition of the premises, we leverage GenericsKB (Bhakthavatsalam et al., 2020), a knowledge base containing statements about commonsense knowledge. The detailed generation process is in Appendix B.

2.2 Circuit Discovery

Our main objective is to find a circuit $\mathcal{C}$ for syllogistic reasoning in a language model $\mathcal{M}$. A circuit $\mathcal{C}$ can be defined as a subset of the original model $\mathcal{M}$ that is both sufficient and necessary for achieving the original model performance on the syllogistic completion task. In order to identify a circuit, we employ activation patching together with circuit ablation (Meng et al., 2022; Vig et al., 2020).

Activation Patching. This technique involves modifying the activation of targeted components and observing the resulting changes. Our study primarily examines the activation of residual streams and multi-head self-attention at the sequence level to trace the term information flow. Activation patching includes three model runs (Clean, Corrupted and Intervened) alongside a quantification process to measure the effect of the interventions (see Figure 2(1)–(4)). Given a masked syllogistic input $x = [\mathcal{P}_1; \mathcal{P}_2; \mathcal{C} \setminus \{p\}]$ as (s, m_1, m_2, p), which can be read as "All s are m_1. All m_2 are p. Therefore, all s are [mask]", and its correct completion $y = p$, the activation patching technique consists of the following steps:

(1) Clean Run. For the target syllogistic input (s, m_1, m_2, p), we record the baseline logit difference, $\delta_+(p, m)$ from the forward pass output of the model (Figure 2(1)).

(2) Corrupted Run. We re-run the model on a corrupted input after applying a specific intervention (e.g., changing the middle term m_2) and record the logit difference $\delta_-(p, m)$ (Figure 2(3)).

(3) Intervened Run. We run the model with the corrupted inputs again and replace the corrupted activations with those from the clean runs to compute the response from the remaining components and measure the causal impact of the intervention. Here, we measure the adjusted logit difference $\delta_p(p, m)$ (Figure 2(2)).

(4) Quantification. We quantify the causal impact of each intervention using a patching score $\mathcal{S}$ following (Heimersheim and Nanda, 2024) as shown in Figure 2(4). This score is further normalized to $[-1, 1]$.

We complement Activation Patching with known methods for analysing hidden activations in transformers, including Logit Lens (Nostalgebraist, 2020) with an input-agnostic approach (Dar et al., 2023; Hanna et al., 2024). Additional technical details can be found in Appendix C.

2.3 Causal Interventions

The choice of tokens to corrupt is a critical aspect of the experimental design, and it is essential to establish an appropriate type of intervention that aligns with the hypothesis being tested. In this study, we employ two distinct interventions to isolate the mechanisms related to the reasoning schema from the propagation of specific token information:

(1) Middle-Term Corruption. To investigate the reasoning mechanism employed in the syllogism completion task, we primarily focus on the transitive property of the middle term, $(m_1 \rightarrow m_2) \wedge (m_1 \equiv m_2)$. We hypothesize that disrupting the transitive property will localize the component responsible for syllogistic reasoning. Therefore, our intervention method replaces the second middle term m_2 with an unseen symbol m_2', breaking the equality and effectively corrupting the validity of the reasoning, pivoting the correct answer towards m: $(s, m_1, m_2, p) \rightarrow (s, m_1, m_2', p)$.

(2) All-Term Corruption. We examine how term-related information flows to the last position for the final prediction to identify potential mover heads. To this end, we replace the original terms (s, m_1, m_2, p) with different terms (s', m_1', m_2', p') while keeping the answers unchanged. We hypothesize that if heads carry the information preferring p over m for the valid prediction, the logit difference will increase; otherwise, it will decrease.

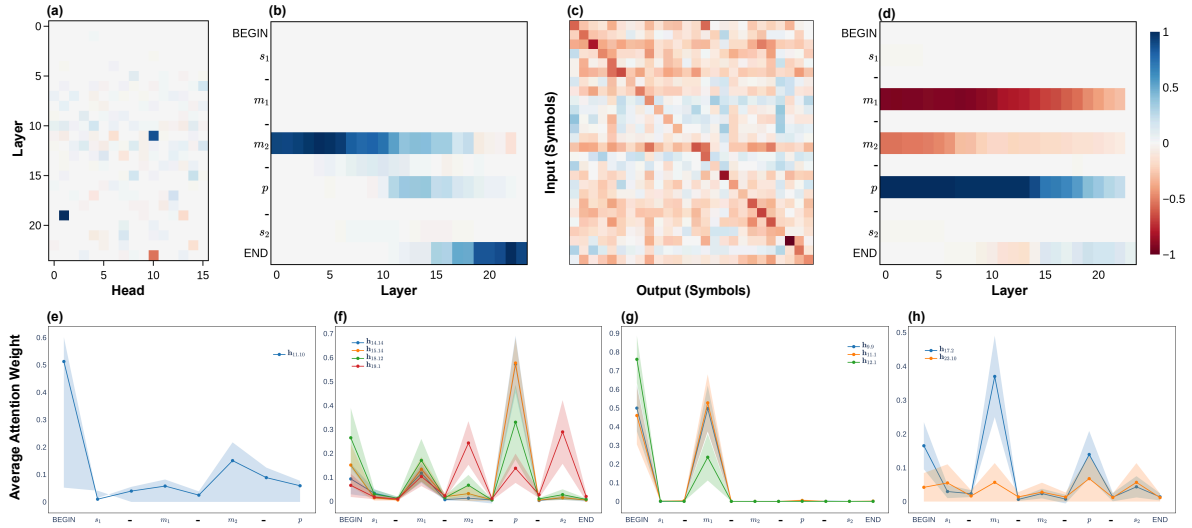

Figure 3: Visualization of the symbolic reasoning circuit identified in the model. (a) Attention output patching and (b) residual stream patching using middle-term interventions highlight the role of attention head $\mathbf{h}_{11.10}$ in aggregating information from the duplicated middle term to the predicate token position $[p]$. (c) Logit lens visualization of the OV (output vector) of $\mathbf{h}_{11.10}$ across 26 uppercase letters reveals its suppressive effect on the attended token at $[p]$. (d) Residual stream patching with all-term corruption indicates that key reasoning information is concentrated at term token positions. (e)–(h) Average attention weights across the batch: (e) attention from token position $[p]$, attending strongly to $[m_2]$; (f)–(h) attention from the last token position, identifying which heads (mover heads) propagate term information to the output. For clarity, a dash (–) on the axis indicates the averaged values for tokens appearing between terms.

2.4 Circuit Ablation and Evaluation

Once we discover a relevant circuit $\mathcal{C}$, we evaluate its necessity and sufficiency using mean ablation. Specifically, we define a circuit $\mathcal{C}$ as necessary if ablating the identified heads $\mathcal{H}$ in $\mathcal{C}$ and preserving the remaining components in $\mathcal{M}$ decreases the original performance. Conversely, we define a circuit $\mathcal{C}$ as sufficient if preserving only the identified heads $\mathcal{H}$ in $\mathcal{C}$ and ablating all the remaining heads in the original model $\mathcal{M}$ is sufficient to obtain the performance of $\mathcal{M}$. To perform mean ablation, we gradually average the target attention heads from downstream to upstream layers and measure the average logit difference $\delta(p, m)$ to assess the impact of the ablation on $\mathcal{M}$. Additional details are included in Appendix D.

3 Mechanistic Interpretation

Empirical Setup. We select the LM for circuit analysis based on a trade-off between model size and performance. To this end, we measure the accuracy of GPT-2 (Radford et al., 2019) at various sizes (117M, 345M, 762M, and 1.5B) on the syllogism completion task, aiming to identify potential phase transition points where performance changes significantly (Jason et al., 2022; Kaplan et al., 2020). We observe a marked transition from small to medium sizes, where average performance across three datasets increases by 439.06% (see Figure 6(a), Appendix E). The shift is even more pronounced under the logit difference metric (see Section 4.4 and Figure 6(b), Appendix E). Based on these findings, we select *GPT-2 Medium*—whose architecture is summarized in Appendix F—for circuit discovery, and subsequently evaluate generalization across model sizes and architectures (Sections 4.3 and 4.4).

3.1 Transitive Reasoning Mechanisms

Middle-term corruption reveals information flow relevant to the transitive property. Figure 3(a-b) presents the results of the middle-term intervention targeting attention heads and residual stream. Figure 3(a), in particular, reveals the positive role of heads $\mathbf{h}_{11.10}$ and $\mathbf{h}_{19.1}$ (where $\mathbf{h}_{l.k}$ denotes the kth head in layer l). Moreover, we observe an information propagation pattern from the $[m_2]$ position to the $[p]$ position (Figure 3(b)). These observations allow us to hypothesize that information from $[m_2]$ is conveyed to $[p]$ on the residual stream by attention head $\mathbf{h}_{11.10}$, which exhibits a strong patching score around the layer responsible

for the propagation. To verify this hypothesis, we further investigate the attention weights between $[p]$ and $[m_2]$. As expected, $[m_2]$ is the position most highly attended by $[p]$, with an average attention weights of 0.15 ± 0.07 (Figure 3(e)). These results suggest that information is indeed moved from $[m_2]$ to $[p]$ on the residual stream subspace, with $\mathbf{h}_{11.10}$ playing a crucial role in this mechanism.

Duplicate middle-term information is aggregated via induction heads. We further investigate how information from $[m_2]$ is moved to $[p]$, positing that this relates to the model's internal reasoning mechanism. We notice that at position $[p]$, the model can observe the complete AAA-1 syllogistic structure (Figure 2(a)) with middle-term duplication. We hypothesize that this structural information for reasoning is collected at one position, given that information refinement occurs at the specific position such as last token (Hanna et al., 2024; Stolfo et al., 2023). To verify this, we employ path patching (Wang et al., 2023), a more selective variant of activation patching, to trace the information flow of head $\mathbf{h}_{11.10}$. Our results show that $\mathbf{h}_{11.10}$ operates based on several induction heads (Elhage et al., 2021) ($\mathbf{h}_{5.8}$, $\mathbf{h}_{6.1}$, $\mathbf{h}_{6.15}$ and $\mathbf{h}_{7.2}$) formed at $[m_2]$. These heads attend to the $[[m_1] + 1]$ token due to the $m_1 \equiv m_2$ conditioned matching operation, likely containing m_1-related information from previous token heads. We conclude, therefore, that m_1 and m_2 information are aggregated at position $[p]$. Additional details of the path patching results are available in the Appendix G.

A suppressive mechanism is revealed through Logit Lens. To better understand the internal mechanism occurring at the $[p]$ position, we investigate attention head $\mathbf{h}_{11.10}$. Having previously examined the attention pattern (composed of attention weights), we now focus on the attention value and output by analyzing the OV circuit ($W_V W_O$) (Elhage et al., 2021) using the logit lens method (Nostalgebraist, 2020). Interestingly, the result in Figure 3(c) shows a clear negative diagonal pattern suggesting that $\mathbf{h}_{11.10}$ strongly suppresses the logit when attending to the same token as the corresponding output. Given our previous findings that $\mathbf{h}_{11.10}$ reads information from the subspace of token $[m_2]$, we conclude that it applies a suppressive mechanism to m-related information and writes it back to the residual stream's subspace at $[p]$. If later heads carry this information to the last token, this mechanism becomes crucial for the model to arrive at the correct answer. For simplicity, we name head $\mathbf{h}_{11.10}$ as m-suppression head.

3.2 Term-Related Information Flow

Key information is moved from term-specific positions to the last position. Now, we use the all-term intervention method to localize mover heads that carry term-related information. In the residual stream patching results (Figure 3(d)), we observe the highest positive score at the $[p]$ position, while negative scores are most prominent at the $[m_1]$ and $[m_2]$ positions, indicating that the information residing in these positions indeed contributes to their corresponding token prediction. This observation aligns with the importance of token embedding information at their respective positions given iterative refinement on the residual stream (Simoulin and Crabbé, 2021). A closer examination of the last token position reveals that the negative effect propagates from relatively early layers (approximately layer 10 onwards), yet positive effects are from later layers (approximately layer 14 onwards), incurring a positive shift of the model's prediction from m to p.

Information is carried by later positive and negative mover heads. Given the findings that the information for prediction resides in term-specific positions, we trace which attention heads transfer the information from each term position ($[p]$, $[m_1]$ and $[m_2]$) to the last token. We call these heads mover heads as the existence of "positive or negative mover heads" that carry or suppress information from the specific token position to the last token position (Wang et al., 2023; García-Carrasco et al., 2024). To identify the sources of information flow, we apply attention value-based patching and isolate nine notable mover heads (see Appendix H for localization details). Among them, $\mathbf{h}_{14.14}$, $\mathbf{h}_{15.14}$, and $\mathbf{h}_{18.12}$ act as positive copy heads, attending strongly to $[p]$, while $\mathbf{h}_{19.1}$ functions as a positive suppression head, exhibiting high attention to $[m_2]$ and $[s]$ (see Figure 3(f)). In contrast, $\mathbf{h}_{9.9}$, $\mathbf{h}_{11.1}$, $\mathbf{h}_{12.1}$, $\mathbf{h}_{17.2}$, and $\mathbf{h}_{23.10}$ serve as negative copy heads, focusing primarily on $[m_1]$ (Figure 3(g)) or showing more diffuse pattern (Figure 3(h)).

A long induction mechanism in early layer movers causes biases towards the middle term. Notably, the early negative mover heads ($\mathbf{h}_{9.9}$, $\mathbf{h}_{11.11}$ and $\mathbf{h}_{12.1}$ in Figure 3(g)) exhibit significantly high attention to $[m_1]$. This pattern indicates

that these heads transfer information from the $[m_1]$ position directly to the last token position, negatively affecting the final prediction and biasing the model towards m. We notice that this behavior closely resembles that of induction heads (Elhage et al., 2021) defined on bi-grams ($[s]$[are]) rather than uni-gram ([are]), supported by previous token heads in upstream layers (e.g., $\mathbf{h}_{8.1}$).

3.3 Circuit Discovery Summary

Overall, we found that the mechanisms for syllogistic inference involve the following phases:

(1) Long Induction: Early layers exhibit biases towards the wrong conclusion due to long-range repetition of the first premise "All A are B".

(2) Duplication: Induction heads aggregate information about duplicated middle terms in the premises.

(3) Suppression: The model aggregates and inhibits middle-term information (i.e.,'B') suppressing the long induction mechanism.

(4) Mover: Token-specific information is propagated to the last position. The process ends in the prediction shift from 'B' to the correct token 'C'.

These results suggest that the circuit is characterized by an internal error correction mechanism. Interestingly, this mechanism is different from the way human experts would reason on syllogistic arguments through the systematic application of abstract logical primitives and inference rules.

4 Circuit Evaluation

The circuit is sufficient and necessary for symbolic arguments. To evaluate the comprehensive correctness of the circuit, we assess necessity and sufficiency via the ablation method described in section 2. The ablation study in Figure 4(a) shows that the identified circuit is both necessary and sufficient, demonstrating a consistent performance degradation when removing circuit components and revealing a complete restoration of the original model's performance when considering only the circuit's subcomponents.

The circuit is robust to superficial variations. We further verify the robustness of the symbolic circuit to superficial and semantic-preserving perturbations. In particular, we modify the letters into numbers (Figure 4(b)) and adopt semantically equivalent quantifiers and related verbs (e.g., "All ... are" is converted into "Each ... is"). The ablation result demonstrates that both types of perturbations do not undermine the sufficiency and necessity property of the circuit.

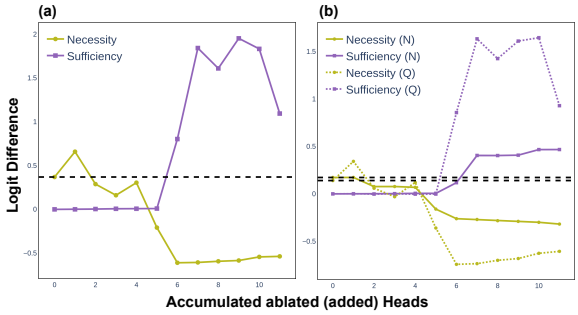

Figure 4: Circuit ablation results for (a) correctness and (b) robustness on the symbolic dataset. Solid lines show numeric perturbation-based performance, dotted lines indicate quantifier perturbation-based performance in (b), and the dashed line shows the baseline logit difference without knockouts. These results validate the necessity, sufficiency, and robustness of the identified symbolic reasoning circuit.

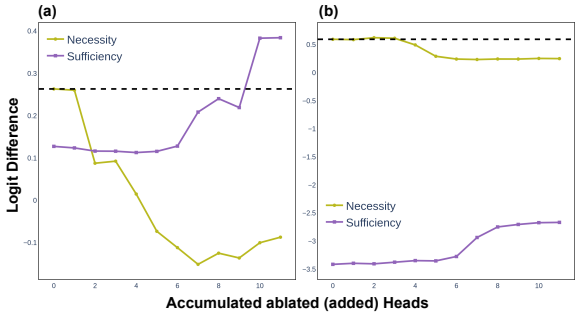

Figure 5: Circuit ablation results for evaluating transferability on non-symbolic datasets: (a) the belief-consistent and (b) the belief-inconsistent. The dashed line represents the baseline logit difference without knockouts. In both cases, the symbolic circuit is necessary, as ablating it reduces performance. However, the circuit is only sufficient for belief-consistent data, as recovery fails in belief-inconsistent settings, suggesting interference from belief-bias.

4.1 Circuit Transferability to Concrete Arguments

The circuit is necessary for non-symbolic arguments, yet not sufficient for belief-inconsistent ones. We present ablation results for the two generated non-symbolic datasets: belief-consistent (Figure 5(a)) and belief-inconsistent (Figure 5(b)). Notably, the symbolic circuit proves necessary for both types of non-symbolic inputs, suggesting that the logic derived from symbolic syllogisms remains

essential even when natural words are substituted. Regarding sufficiency, while belief-consistent data show significant performance recovery, we observe an inability to restore performance on the belief-inconsistent set despite an increasing trend. These results indicate that belief biases encoded in different attention heads may play an important role.

Belief biases corrupt reasoning mechanisms. To further investigate the impact of belief biases, we again conduct intervention experiments. Specifically, we observe that in the AAA-1 syllogism schema (s, m_1, m_2, p), the subject token s should be irrelevant for deriving the correct answer p over m. Therefore, we leverage this property to verify whether word-specific biases are introduced when instantiating the schema with concrete knowledge. To this end, we perform activation patching by corrupting the subject term, transforming (s, m_1, m_2, p) to (s', m_1, m_2, p), and measuring the degradation in logit difference as $\delta_+(p, m) - \delta_-(p, m)$. Notably, the non-symbolic setting exhibits a significant performance degradation of 0.66 ± 1.27, representing a 299.96% change from the baseline. In contrast, the symbolic setting shows a minimal degradation of 0.00 ± 0.64, a mere 0.35% drop. This drastic difference supports our hypothesis that the knowledge acquired during pre-training corrupts the content-independent reasoning circuit identified on the symbolic set with additional attention heads.

4.2 Generalization to Syllogistic Schemes

We further aim to understand whether the symbolic circuit is specific to the AAA-1 syllogism (*Barbara*). To this end, we extend our experiment to encompass all 15 unconditionally valid syllogisms (see Appendix A and I for details about the syllogistic schemes and more experimental results). To evaluate the transferability to all 15 schemes, we verify three main conditions, reported in Table 2: (C1) Necessity, (C2) Sufficiency, and (C3) Positive Logit Difference. We also measure the accuracy of the completion task for each syllogism. From Table 2, we observe that the circuit is sufficient and necessary for all the syllogistic schemes in which the model achieves high downstream accuracy ($\geq$ 60%), supporting the conclusion that the circuit includes components that are crucial for the emergence of syllogistic reasoning in general. Notably, all three conditions are satisfied for the affirmative syllogisms (AII-3, IAI-3, IAI-4), for which the model achieves an accuracy above 0.6.

Mood-Figure	C1	C2	C3	Acc
AII-3 (*Datisi*)	✓	✓	✓	**0.84**
IAI-3 (*Disamis*)	✓	✓	✓	**0.68**
IAI-4 (*Dimaris*)	✓	✓	✓	**0.68**
AAA-1 (*Barbara*)	✓	✓	✓	**0.67**
EAE-1 (*Celarent*)	✓	-	✓	**0.59**
EIO-4 (*Fresison*)	✓	-	✓	**0.53**
EIO-3 (*Ferison*)	✓	-	✓	**0.53**
AII-1 (*Darii*)	✓	✓	-	0.43
AOO-2 (*Baroco*)	-	-	-	0.24
AEE-4 (*Camenes*)	-	-	-	0.24
OAO-3 (*Bocardo*)	✓	-	-	0.22
EIO-1 (*Ferio*)	✓	-	-	0.18
EIO-2 (*Festino*)	-	-	-	0.09
EAE-2 (*Cesare*)	-	-	-	0.08
AEE-2 (*Camestres*)	-	-	-	0.04

Table 2: Generalizability across unconditionally valid syllogistic forms. Columns C1, C2, and C3 indicate whether conditions are met (✓) or not (-): C1: Necessity, C2: Sufficiency, C3: Positive Logit Difference. 'Accuracy (Acc)' shows performance on the completion task for each syllogism. The results show that the model achieves high downstream accuracy particularly for affirmative syllogisms.

4.3 Generalization to Model Sizes

Next, we expand our analysis to different sizes of GPT-2 (small, large and XL). Here, the intervention results show similar suppression mechanism patterns and information flow across all models. However, the residual stream of GPT-2 small in the all-term corruption setup is reversed due to its low downstream accuracy (see Appendix J for additional details). Moreover, as the model size increases, we found evidence that the contribution of attention heads becomes more complex. We hypothesize this might be caused by a stronger impact of world knowledge with increasing size, as suggested by the decrease in accuracy on the symbolic dataset and the increase on the non-symbolic set observed for the XL model (see Figure 6(a)–(b), Appendix E).

4.4 Generalization to Model Families

We further extend our investigation to assess whether a similar reasoning mechanism can be identified across different LMs. Specifically, we evaluate the circuit on: Pythia with 70M, 160M, 410M, and 1B parameters (Biderman et al., 2023); Qwen 2.5 with 0.5B and 1.5B parameters (Yang et al., 2024); Llama 3.2 with 1B parameters (et al., 2024); and additionally, GPT-2-medium after be-

Model	S	BC	BI
Group 1: Erratic Pattern			
Pythia-70M	0.19	0.02	0.02
Pythia-160M	0.00	0.12	0.21
Group 2: Compatible Pattern			
Pythia-410M	0.99	0.40	0.51
Pythia-1B	0.30	0.31	0.43
Llama-3.2-1B	1.00	0.81	0.70
Group 3: Variant Pattern			
Qwen2.5-0.5B	1.00	0.96	0.90
Qwen2.5-1.5B	1.00	0.99	0.90

Table 3: Accuracy comparison across different models: **symbolic (S)**, **belief-consistent (BC)**, and **inconsistent (BI)**. **Group 1** models exhibit erratic patterns, failing to establish a stable reasoning mechanism. **Group 2** models exhibit a compatible reasoning circuit with consistent information flow and suppression. **Group 3** exhibits a functional but distinct suppression mechanism, where suppression is applied at the last token position.

ing fine-tuned on argumentative texts for critical thinking (Betz et al., 2021).

Our results reveal distinct grouping patterns. In relatively small models—such as Pythia 70M and 160M—we observe erratic activation patterns that mirror the limitations seen in under-scaled models like GPT-2 small. In contrast, models with more than 0.5B parameters (namely, Pythia 410M, Pythia 1B, and Llama 3.2 1B) exhibit a compatible pattern in terms of information flow and suppressive mechanisms. Qwen exhibits a variant suppressive mechanism, with the suppression head operating at the last token position rather than $[p]$. Meanwhile, the fine-tuned GPT-2 medium model yields activation patterns that are nearly indistinguishable from its pre-trained counterpart, suggesting that the core reasoning circuit originates from pre-training rather than fine-tuning. Table 3 summarizes the accuracy of each model across the Symbolic, Belief-Consistent, and Belief-Inconsistent datasets (see further circuit analysis details in Appendix K).

Overall, our results indicate that while the reasoning circuit generalizes effectively across sufficiently scaled models, subtle architectural nuances can lead to distinct but comparable dynamics.

5 Related Work

Mechanistic circuit analysis has emerged as a promising approach to interpreting the internal mechanisms of Transformers (Olah et al., 2020; Nanda et al., 2023; Olsson et al., 2022; Wang et al., 2023; García-Carrasco et al., 2024; Hanna et al., 2024). Existing approaches investigating internal reasoning mechanisms mainly focus on math-related tasks, elucidating the information flow for answering mathematical questions (Stolfo et al., 2023) and examining arithmetic operations (Quirke and Barez, 2024). Most pertinent to our work, Wiegreffe et al. (2025) provides a mechanistic interpretation of multiple-choice question answering, investigating attention head-level patterns. In general, our mechanistic analysis complements recent work investigating the challenges in processing reasoning arguments that contradict established beliefs and whether the reasoning in LMs stems from internalized rules or memorized content (Ando et al., 2023; Yu et al., 2023; Wu et al., 2024; Talmor et al., 2020; Wu et al., 2024; Kassner et al., 2020; Haviv et al., 2023; Feldman, 2020; Monea et al., 2024; Yu et al., 2023; Wallat et al., 2020; Eisape et al., 2024). To address this question, different mechanistic approaches have been adopted to localizing factual associations (Meng et al., 2022; Geva et al., 2023; Dai et al., 2022) or assessing conditions for generalization (Wang et al., 2024). However, to the best of our knowledge, we are the first to investigate mechanisms for syllogistic reasoning.

6 Conclusion

In this study, we presented a comprehensive mechanistic interpretation of syllogistic reasoning in language models. By combining activation patching, embedding space analysis, and circuit ablation, we uncovered a structured error-correction mechanism characterized by the suppression of duplicated middle terms and the propagation of relevant information via mover heads. Our findings show that this circuit is necessary and sufficient for symbolic syllogistic reasoning. Moreover, we demonstrated that, while this reasoning mechanism generalizes across syllogistic schemes, model architectures and sizes, it remains susceptible to contamination from belief biases when instantiated with natural language inputs. Overall, our findings suggest that LMs learn transferable reasoning mechanisms but that, at the same time, such mechanisms might be contaminated and suppressed by the same world knowledge acquired during pre-training. Further studies will be required to understand how the discovered symbolic circuit can be effectively disentangled from world knowledge to enable systematic generalization in LMs' reasoning.

7 Limitations

It is important to acknowledge some of the limitations of our study. First, our analysis is conducted predominantly on the transitive property and term-specific information and does not consider the full spectrum of the reasoning dynamics that might appear in more complex scenarios which are hard to model via causal intervention techniques. Second, Our experimentation is confined to specific syllogistic reasoning templates due to the intricate level of granularity and the computational complexity required in circuit analysis. Despite these limitations, however, we believe our findings could offer valuable insights into the reasoning mechanisms adopted by language models for formal reasoning and related biases, laying a solid foundation for future research in this field.

Acknowledgements

This work was funded by the Swiss National Science Foundation (SNSF) project "NeuMath" (200021_204617), by the CRUK National Biomarker Centre, and supported by the Manchester Experimental Cancer Medicine Centre and the NIHR Manchester Biomedical Research Centre.

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. In *Proceedings of the 4th Natural Logic Meets Machine Learning Workshop*, pages 1–11. Association for Computational Linguistics.

Leonardo Bertolazzi, Albert Gatt, and Raffaella Bernardi. 2024. A systematic analysis of large language models as soft reasoners: The case of syllogistic inferences. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13882–13905, Miami, Florida, USA. Association for Computational Linguistics.

Gregor Betz, Christian Voigt, and Kyle Richardson. 2021. Critical thinking for language models. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 63–75, Groningen, The Netherlands (online). Association for Computational Linguistics.

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. Genericskb: A knowledge base of generic statements. *Preprint*, arXiv:2005.00660.

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

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

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *Preprint*, arXiv:2303.12712.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3882–3890. International Joint Conferences on Artificial Intelligence Organization. Main track.

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

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

Tiwalayo Eisape, Michael Tessler, Ishita Dasgupta, Fei Sha, Sjoerd Steenkiste, and Tal Linzen. 2024. A systematic comparison of syllogistic reasoning in humans and language models. 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 8425–8444, Mexico City, Mexico. Association for Computational Linguistics.

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

Aaron Grattafiori et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jack W. Rae et al. 2022. Scaling language models: Methods, analysis & insights from training gopher. *Preprint*, arXiv:2112.11446.

Vitaly Feldman. 2020. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 954–959, New York, NY, USA. Association for Computing Machinery.

Jorge García-Carrasco, Alejandro Maté, and Juan Carlos Trujillo. 2024. How does GPT-2 predict acronyms? extracting and understanding a circuit via mechanistic interpretability. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3322–3330. PMLR.

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

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

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2024. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 248–264, Dubrovnik, Croatia. Association for Computational Linguistics.

Stefan Heimersheim and Neel Nanda. 2024. How to use and interpret activation patching. *Preprint*, arXiv:2404.15255.

Keith J Holyoak and Robert G Morrison. 2005. *The Cambridge handbook of thinking and reasoning*. Cambridge University Press.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online. Association for Computational Linguistics.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.

Andrew K Lampinen, Ishita Dasgupta, Stephanie CY Chan, Hannah R Sheahan, Antonia Creswell, Dharshan Kumaran, James L McClelland, and Felix Hill. 2024. Language models, like humans, show content effects on reasoning tasks. *PNAS nexus*, 3(7):pgae233.

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. 2015. Convergent learning: Do different neural networks learn the same representations? In *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, volume 44 of *Proceedings of Machine Learning Research*, pages 196–212, Montreal, Canada. PMLR.

Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kiciman, Hamid Palangi, Barun Patra, and Robert West. 2024. A glitch in the matrix? locating and detecting language model grounding with fakepedia. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6828–6844, Bangkok, Thailand. Association for Computational Linguistics.

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

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

Nostalgebraist. 2020. Interpreting gpt: The logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed: 2024-08-11.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024.001.

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

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Philip Quirke and Fazl Barez. 2024. Understanding addition in transformers. In *The Twelfth International Conference on Learning Representations*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

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

Julia Rozanova, Deborah Ferreira, Mokanarangan Thayaparan, Marco Valentino, and Andre Freitas. 2022. Decomposing natural logic inferences for neural NLI. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 394–403, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Julia Rozanova, Marco Valentino, Lucas Cordeiro, and André Freitas. 2023. Interventional probing in high dimensions: An NLI case study. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2489–2500, Dubrovnik, Croatia. Association for Computational Linguistics.

Julia Rozanova, Marco Valentino, and André Freitas. 2024. Estimating the causal effects of natural logic features in transformer-based NLI models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6319–6329, Torino, Italia. ELRA and ICCL.

Antoine Simoulin and Benoit Crabbé. 2021. How many layers and why? an analysis of the model depth in transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 221–228.

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

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Marco Valentino, Geonhee Kim, Dhairya Dalal, Zhixue Zhao, and André Freitas. 2025. Mitigating content effects on reasoning in language models through fine-grained activation steering. *arXiv preprint arXiv:2505.12189*.

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.

Jonas Wallat, Jaspreet Singh, and Avishek Anand. 2020. BERTnesia: Investigating the capture and forgetting of knowledge in BERT. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 174–183, Online. 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*.

Zhiwei Wang, Yunji Wang, Zhongwang Zhang, Zhangchen Zhou, Hui Jin, Tianyang Hu, Jiacheng Sun, Zhenguo Li, Yaoyu Zhang, and Zhi-Qin John Xu. 2024. Towards understanding how transformer perform multi-step reasoning with matching operation. *CoRR*, abs/2405.15302.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

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*.

Yongkang Wu, Meng Han, Yutao Zhu, Lei Li, Xinyu Zhang, Ruofei Lai, Xiaoguang Li, Yuanhang Ren, Zhicheng Dou, and Zhao Cao. 2023. Hence, socrates is mortal: A benchmark for natural language syllogistic reasoning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2347–2367, Toronto, Canada. Association for Computational Linguistics.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2024. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. 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 1819–1862, Mexico City, Mexico. Association for Computational Linguistics.

Magdalena Wysocka, Danilo Carvalho, Oskar Wysocki, Marco Valentino, and Andre Freitas. 2025. SylloBio-NLI: Evaluating large language models on biomedical syllogistic reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7235–7258, Albuquerque, New Mexico. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105–6117, Online. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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

A Unconditionally Valid Syllogism Schemes

We list all 15 unconditionally valid syllogisms (Table 4).

B Dataset Generation

B.1 Symbolic

The symbolic dataset comprises sentences where all terms in the premises are represented by abstract symbols (uppercase alphabet letters). From the set of all 26 uppercase alphabets, three-letter triples (e.g., A, B and C) are randomly sampled. For each triple, six permutated prompts and label pairs are generated following syllogism templates designed to minimise latent semantic interference among alphabet symbol tokens.

B.2 Non-symbolic

The non-symbolic datasets are constructed based on GenericsKB (Bhakthavatsalam et al., 2020), a resource that provides a foundation for the evaluation of sentence veracity with associated truthfulness scores (0-1). The dataset construction process involves the following steps:

- **Extraction:** Select generic sentences with a truthfulness score of 1 based on GenericsKB (Bhakthavatsalam et al., 2020), specifically those in the form *A are B*, using regular expression.
- **Syllogism Construction:** We form universal affirmative syllogism arguments (*Barbara*) based on a template by chaining sentences where the predicate of one sentence logically matches the subject of another.
- **Constraints:** We exclude terms tokenized into multiple tokens to maintain consistency in comparison with the symbolic dataset, which is essential for coherent circuit analysis.
- **Classification:** To address issues of partial inclusion and syntactic ambiguity in constructed syllogism arguments, we employ GPT-4 (OpenAI, 2024) to classify whether arguments contain only truthful premises and whether the middle-terms are syntactically and semantically equivalent. This step helps us classify a belief-consistent dataset and a belief-inconsistent dataset.

Name	Mood	Figure	Premise (m)	Premise (M)	Conclusion
Barbara	AAA	1	$\forall x(s, m)$	$\forall x(m, p)$	$\forall x(s, p)$
			All A are B.	*All B are C.*	*All A are C.*
Celarent	EAE	1	$\forall x(s, m)$	$\forall x\neg(m, p)$	$\forall x\neg(s, p)$
			All A are B.	*No B are C.*	*No A are C.*
Darii	AII	1	$\exists x(s, m)$	$\forall x(m, p)$	$\exists x(s, p)$
			Some A are B.	*All B are C.*	*Some A are C.*
Ferio	EIO	1	$\exists x(s, m)$	$\forall x\neg(m, p)$	$\exists x\neg(s, p)$
			Some A are B.	*No B are C.*	*Some A are not C.*
Camestres	AEE	2	$\forall x\neg(s, m)$	$\forall x(p, m)$	$\forall x\neg(s, p)$
			No A are B.	*All C are B.*	*No A are C.*
Cesare	EAE	2	$\forall x(s, m)$	$\forall x\neg(p, m)$	$\forall x\neg(s, p)$
			All A are B.	*No C are B.*	*No A are C.*
Baroco	AOO	2	$\exists x\neg(s, m)$	$\forall x(p, m)$	$\exists x\neg(s, p)$
			Some A are not B.	*All C are B.*	*Some A are not C.*
Festino	EIO	2	$\exists x(s, m)$	$\forall x\neg(p, m)$	$\exists x\neg(s, p)$
			Some A are B.	*No C are B.*	*Some A are not C.*
Disamis	IAI	3	$\forall x(m, s)$	$\exists x(m, p)$	$\exists x(s, p)$
			All B are A.	*Some B are C.*	*Some A are C.*
Datisi	AII	3	$\exists x(m, s)$	$\forall x(m, p)$	$\exists x(s, p)$
			Some B are A.	*All B are C.*	*Some A are C.*
Ferison	EIO	3	$\exists x(m, s)$	$\forall x\neg(m, p)$	$\exists x\neg(s, p)$
			Some B are A.	*No B are C.*	*Some A are not C.*
Bokardo	OAO	3	$\forall x(m, s)$	$\exists x\neg(m, p)$	$\exists x\neg(s, p)$
			All B are A.	*Some B are not C.*	*Some A are not C.*
Dimaris	IAI	4	$\forall x(m, s)$	$\exists x(p, m)$	$\exists x(s, p)$
			All B are A.	*Some C are B.*	*Some A are C.*
Camenes	AEE	4	$\forall x\neg(m, s)$	$\forall x(p, m)$	$\forall x\neg(s, p)$
			No B are A.	*All C are B.*	*No A are C.*
Fresison	EIO	4	$\exists x(m, s)$	$\forall x\neg(p, m)$	$\exists x\neg(s, p)$
			Some B are A.	*No C are B.*	*Some A are not C.*

Table 4: 15 Unconditionally valid syllogism schemes. The table lists the syllogisms by their traditional names, moods, and figures, with formalized logical expressions on the first line and corresponding natural language examples on the second line. The minor premise (m) is presented before the major premise (M) to emphasize the transitive property in AAA-1 syllogism.

- **Validation:** We manually evaluate the alignment of classified arguments with human belief-consistency (i.e., premises are true and logic is valid).

This process ensures that our non-symbolic dataset maintains logical equivalence with the symbolic dataset while incorporating meaningful semantic real-word concepts.

B.3 Data Statistics

For all experiments, we use 90 samples each for both symbolic and non-symbolic arguments to balance analytical depth with computational efficiency. This sample size sufficiently yields statistically significant activation patching results ($p < 0.05$). We organize data statistics for generated datasets (Table 5).

C Embedding Space Analysis Details

One established method for analyzing hidden activation in transformer language models is the logit lens, which projects activation into embedding space (Nostalgebraist, 2020; Elhage et al., 2021; Geva et al., 2022). We focus on an input-agnostic approach (Dar et al., 2023; Hanna et al.,

Statistic	SYM	BC	BI
Number of Samples	90	90	90
Token Sequence Length	15	15	15
Unique Terms (s)	26	87	86
Unique Terms (m)	26	70	35
Unique Terms (p)	26	59	64

Table 5: Summary of Dataset Statistics. **SYM** refers to the symbolic dataset, **BC** refers to the non-symbolic belief-consistent dataset, and **BI** refers to the non-symbolic belief-inconsistent dataset.

2024), utilizing both unembedding (W_U) and embedding (W_E) matrices to construct a $\mathbb{R}^{|V|\times|V|}$ matrix, where $|V|$ denotes the model's vocabulary size. This study employs the OV circuit (Elhage et al., 2021) of attention heads, formed by attention value and output weights ($W_V W_O$), to understand how source tokens generally influence output logits. The OV circuit-based logit lens for attention head $\mathbf{h}$ is formalized as $W_E W_V^{\mathbf{h}} W_O^{\mathbf{h}} W_U \in \mathbb{R}^{|V|\times|V|}$. Following the previous works (Elhage et al., 2021; Dar et al., 2023), this formulation omits layer normalization.

D Circuit Ablation Method Details

To measure the necessity and sufficiency of the circuit $\mathcal{C}$ in the model $\mathcal{M}$, we knock out attention heads in $\mathcal{H}$ from the model $\mathcal{M}$ and measure the average logit difference $\delta(p, m)$ along the batch. We denote the logit difference in circuit state $\mathcal{C}$ as $\delta(p, m, \mathcal{C})$ and every subset of heads set as $H \subset \mathcal{H}$.

In order to verify the head's necessity in the model, we conduct a cumulative ablation of $\mathcal{C}$ from total circuit $\mathcal{M}$, progressing from downstream to upstream layers. At each ablation step k, where $\mathcal{C}_k^* = \mathcal{M} \setminus H_k$ and H_k denotes the set of ablated attention heads up to step k as:

$$\mathbb{E}_{x\sim X}[\delta_k(p, m, \mathcal{C}_k^*)] \quad \text{where} \quad \mathcal{C}_k^* = \mathcal{M} \setminus H_k$$

Conversely, we perform a cumulative addition of attention heads for evaluating the sufficiency of the circuit, starting from earlier layers and progressing to later ones, while maintaining all other attention heads in a mean-ablated state. At each addition step j, where $\mathcal{C}_j^* = \mathcal{M} \setminus (\mathcal{H} \setminus H_j)$ and H_j denotes the set of added attention heads up to step j as:

$$\mathbb{E}_{x\sim X}[\delta_j(p, m, \mathcal{C}_j^*)] \quad \text{where} \quad \mathcal{C}_j^* = \mathcal{M} \setminus (\mathcal{H} \setminus H_j)$$

E Phase Transition

In Figure 6(a–b), we observe a notable phase transition from small to medium models in accuracy and logit difference.

F GPT-2 Model Architecture

We provide a self-contained concise overview of the GPT-2 model architecture, highlighting the main components and their mathematical relationships. Bias terms are not presented for the simplicity. We refer the conventions of notation in Elhage et al. (2021) and Geva et al. (2023).

Notation

- x_i - One-hot encoded vector representing the i-th token in the input sequence.
- p_i - One-hot encoded vector representing the i-th positional information.
- r_i^l - Residual stream vector representing the i-th token at layer l in the input sequence.
- $W_E^{\text{token}}, W_E^{\text{pos}}$ - Token and positional embedding matrices.
- A_i^l - Output from the Multi-Head Self-Attention sublayer
- M_i^l - Output from Multi-Layer Perceptron (MLP) sublayer.
- W_Q, W_K, W_V, W_O - Query, key, value, output weight matrices of attention heads.
- $W^{\text{in}}, W^{\text{out}}$ - Input and output weight matrices of MLP feed-forward networks.
- d - Dimension of the attention head state embedding vectors.
- D - Dimension of the hidden state embedding vectors.
- N - Length of the input sequence.
- L - Total number of transformer layers.
- $\mathcal{V}$ - Vocabulary set of the model.
- σ and σ' - Activation functions used in self-attention and MLP sublayers, respectively.

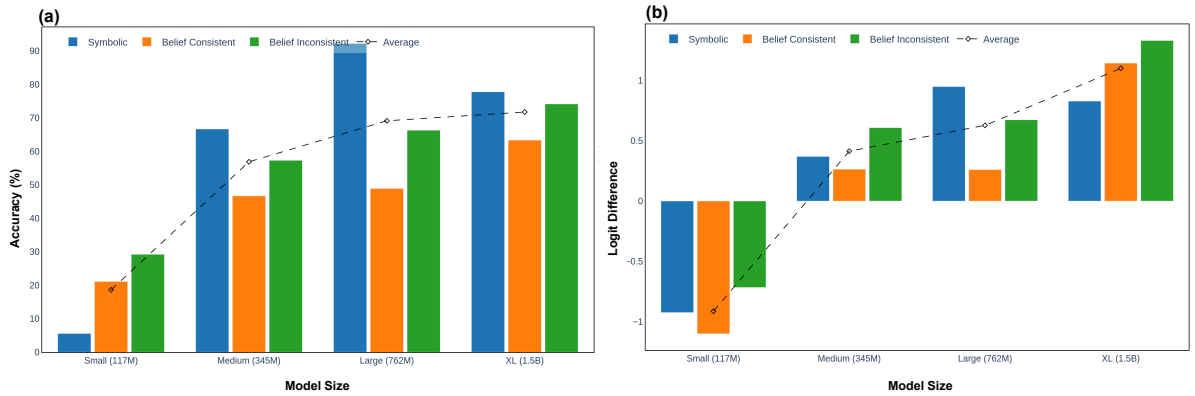

Figure 6: Accuracy and logit difference on the syllogism completion task across all different sizes of GPT-2 for three different datasets.

Embedding Layer. Each token in the input sequence is transformed into an embedded representation by combining token-specific and positional embeddings:

$$r_i^0 = x_i W_E^{\text{token}} + p_i W_E^{\text{pos}}$$

This operation initializes the input for the transformer layers, where r_i^0 represents the initial embedded state of the i-th token.

Residual Stream. The residual stream facilitates the propagation of information across transformer layers. It is updated at each layer by contributions from the multi-head self-attention and multi-layer perceptron sublayers:

$$r_i^l = r_i^{l-1} + A_i^l + M_i^l$$

Multi-Head Self-Attention Sublayer. This sublayer processes information by applying self-attention mechanisms across multiple 'heads' of attention, enabling the model to capture various aspects of the input data, where σ represent the non-linearity function:

$$A_i^l = \sum_{h \in H} \left(\sigma \left(\frac{(r_i^l W_Q^h)(r_i^l W_K^h)^T}{\sqrt{d_K}} \right) (r_i^l W_V^h) \right) W_O^h$$

Each head computes a weighted sum of all tokens' transformed states, focusing on different subsets of sequence information.

Multi-Layer Perceptron Sublayer. Each token's representation is further processed in a position-wise manner by the MLP sublayer:

$$M_i^l = \sigma'(r_i^l W^{in})(W^{out})^T$$

The MLP modifies each token's state locally, enhancing its ability to process information.

Layer Normalization. Before processing by self-attention and MLP processing, each token's state is normalized to stabilize learning and reduce training time:

$$\text{LN}(r_i^l) = \gamma \odot \frac{r_i^l - \mu_i^l}{\sqrt{(\sigma_i^l)^2 + \epsilon}} + \beta$$

This step ensures that the activations across different network layers maintain a consistent scale.

Prediction Head. The prediction head generates logits for the next token prediction using the final transformed states:

$$\text{logits}_N = \text{LN}(r_N^L) W_U$$

The predicted token is chosen based on the sampling method (e.g., greedy decoding) at the last position N.

G Path Patching Details

Path patching, an generalized version of activation patching, aims to compute direct effects among model components rather than indirect effects diffused from intervened components (Wang et al., 2023). This method involves freezing non-targeted activations during an initial forward pass, storing the targeted activation state in intervention, and then executing a subsequent forward pass with the targeted activation state substituted by the stored one. This approach isolates targeted component-to-component effects, eliminating non-relevant interactions. In our implementation, we employ a noising method where clean activations are replaced with corrupted ones, resulting in negative scores indicating positive contributions from corresponding components. Our findings, illustrated in Figure

7, reveal that $\mathbf{h}_{11.10}$ strongly operates based on several heads: $\mathbf{h}_{5.8}$, $\mathbf{h}_{6.1}$, $\mathbf{h}_{6.15}$, and $\mathbf{h}_{7.2}$. Manual investigation of attention patterns subsequently confirmed these as induction heads (Elhage et al., 2021).

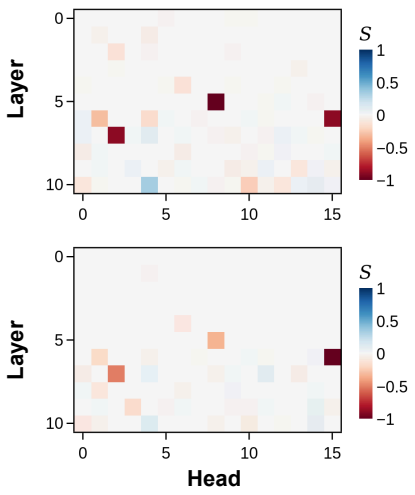

Figure 7: Path patching results for (Top) symbolic inputs and non-symbolic inputs (Bottom) in the middle-term corruption setup. Note that negative heads are positively influencing heads, as we replace corrupted activation with clean ones in our noising method.

H Localizing Mover Heads

To efficiently classify the numerous attention heads (384 in GPT-2 medium) within the model, we calculate the Positional Patching Difference (PPD) to construct a four-quadrant analysis:

$$\text{PPD} = |S[p]| - |S[m_1] + S[m_2]|$$

This method involves intervening in the attention value weights W_V and constructs a distribution layout of heads, providing a systematic classification for quadrant groups. The layout is composed of the PPD score on the x-axis and the patching score for all sequence positions ($\mathcal{S}$) on the y-axis.

Based on this PPD score quadrant layout analysis, we categorize heads according to their quadrant positions, revealing insights into their functional roles:

(1) First quadrant (Positive Copy Candidates): Heads with $\mathcal{S} > 0$ and positive PPD, predominantly copying $[p]$-based information.

(2) Second quadrant (Positive Suppression Candidates): Heads with $\mathcal{S} > 0$ but negative PPD, predominantly suppressing $[m]$-based information.

(3) Third quadrant (Negative Copy Candidates): Heads with $\mathcal{S} < 0$ and negative PPD, predominantly copying $[m]$-based information.

(4) Fourth quadrant (Negative Suppression Candidates): Heads with $\mathcal{S} < 0$ but positive PPD, predominantly suppressing $[p]$-based information.

The intervention of attention head values is selected because all-term corruption maintains the same positional alignment for all samples $(s, m_1, m_2, p) \rightarrow (s', m_1', m_2', p')$, resulting in minimal impact from attention head pattern interventions composed by attention query and key. Consequently, we can infer that the attention head output patching results primarily are derived from the attention value activation. This property enables us to localize the source token positions from which the term information is moved by attention value patching.

Formally, when we denote the attention query, key, value, and output weight matrices as W_Q, W_K, W_V, and W_O respectively, and the residual stream vector as r, one attention head output can be expressed as: $\sigma\left((rW_Q)(rW_K)^T\right)(rW_V)W_O$, where σ represents the non-linearity function (softmax in GPT-2) applied in the attention pattern.

To include only influential heads in the circuit, we extracted outlier heads with an absolute patching score exceeding the threshold (τ), defined by the mean and standard deviation of all heads' scores as:

$$\{\mathbf{h} \mid |\mathcal{S}_\mathbf{h}| > \tau\} \quad \text{where} \quad \tau = \mu + 2\sigma$$

Our analysis identifies 9 heads for the symbolic circuit, as illustrated by red dots in Figure 8(a). $\mathbf{h}_{14.14}$, $\mathbf{h}_{15.14}$, and $\mathbf{h}_{18.12}$ are classified as positive copy candidates, while $\mathbf{h}_{19.1}$ is grouped as a positive suppression candidates. $\mathbf{h}_{9.9}$, $\mathbf{h}_{11.1}$, $\mathbf{h}_{12.1}$, $\mathbf{h}_{17.2}$, and $\mathbf{h}_{23.10}$ are classified as negative copy candidates. It is noteworthy that non-symbolic results reveal a more complex and noisy pattern in the mover localization process (Figure 8(b)).

I Generalizability to Other Syllogisms

To assess the generalizability of our model, we applied the circuit ablation method to all 15 unconditionally valid syllogisms (Table 4), and the result

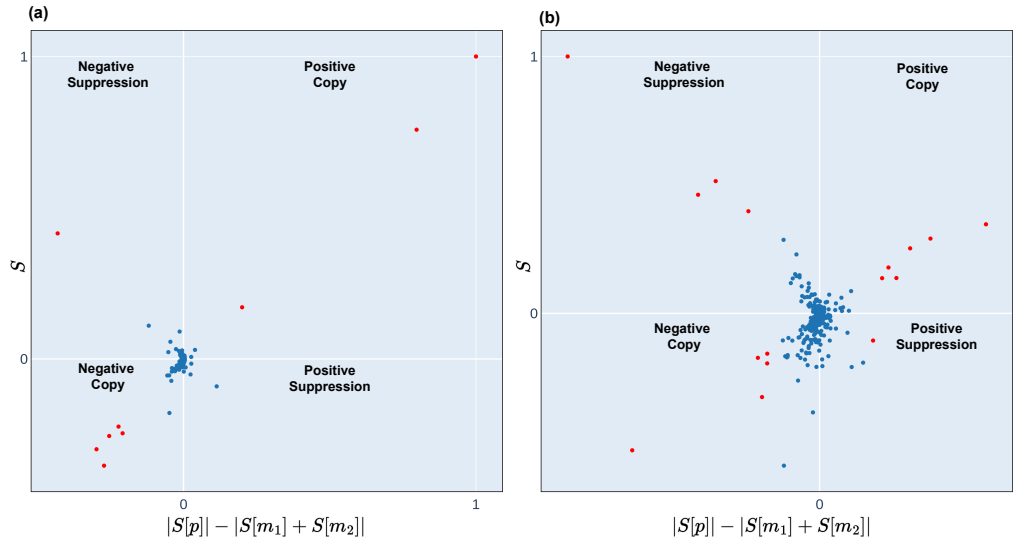

Figure 8: Distribution of attention heads in PPD-based quadrant analysis. (a) Results for symbolic inputs. (b) Results for non-symbolic inputs. The x-axis represents normalized PPD scores, while the y-axis represents normalized attention value-based patching scores (S). Red dots indicate attention heads above the threshold (τ) of the patching score.

is shown in Figure 9. Unconditionally valid syllogisms are logical arguments that maintain their validity irrespective of the truth values of their premises. The validity of these syllogisms is determined solely by their logical form, which is characterized by mood and figure combinations. The mood of a syllogism is defined by the arrangement of four proposition types ("All (A)", "No (E)", "Some (I)", "Some ... not (O)") across its two premises and conclusion. Conversely, the figure of a categorical syllogism is determined by the structural arrangement of terms within the constituent sentences. This approach enables the evaluation of syntactic logic in isolation from contextual interference and belief-bias effects.

J Generalizability Across Model Sizes

Our approach is guided by the universality hypothesis, which suggests that similar representational patterns and circuits emerge across models trained on the same dataset (Olah et al., 2020; Kornblith et al., 2019; Li et al., 2015). We assume that core mechanisms should persist across these different model sizes. Figure 10 presents the results of this comparative analysis.

K Generalizability Across Different Models

Figures 11 and 12 represent the results of applying the same circuit analysis in different model families. The m-suppression heads for each model were manually selected based on their likelihood, determined by patching scores and positioning within the activation pattern.

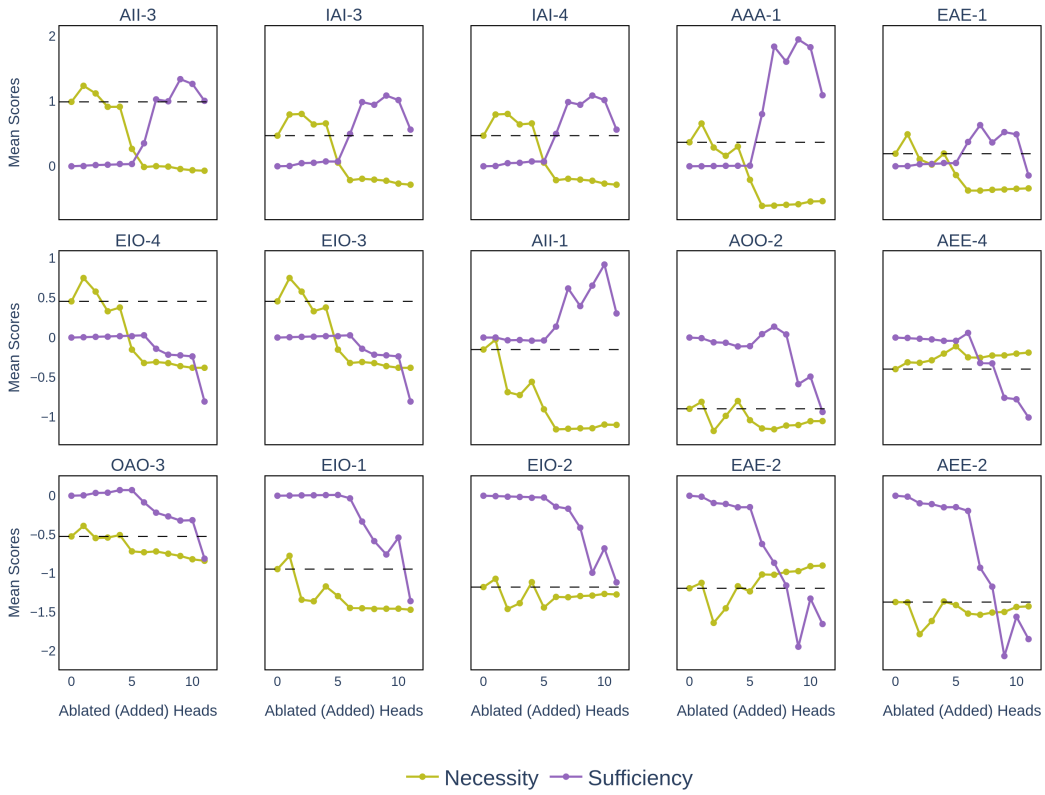

Figure 9: Necessity and sufficiency performance results from circuit ablation method for all unconditionally valid syllogistic forms. Labels (e.g., AII-3) denote mood and figure combinations.

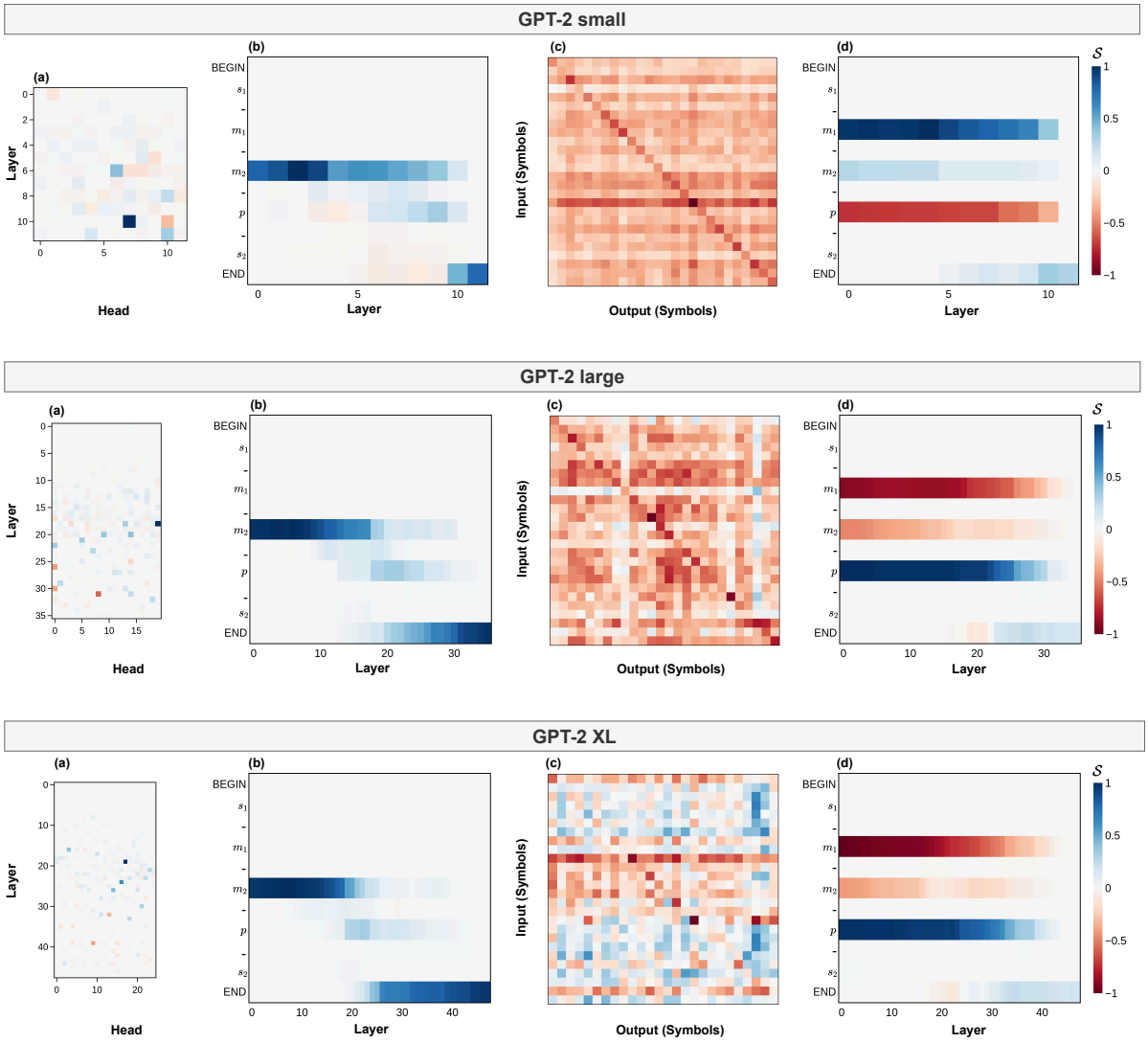

Figure 10: Comprehensive results of the symbolic circuit analysis across different model sizes. (a) Attention output patching results and (b) residual stream patching results in the middle-term intervention setup. (c) OV circuit logit lens results for m-suppression head, with input and output comprising 26 uppercase letters. (d) Residual stream patching results in the all-term corruption setup. For clarity, a dash (–) indicates the averaged values for tokens appearing between terms.

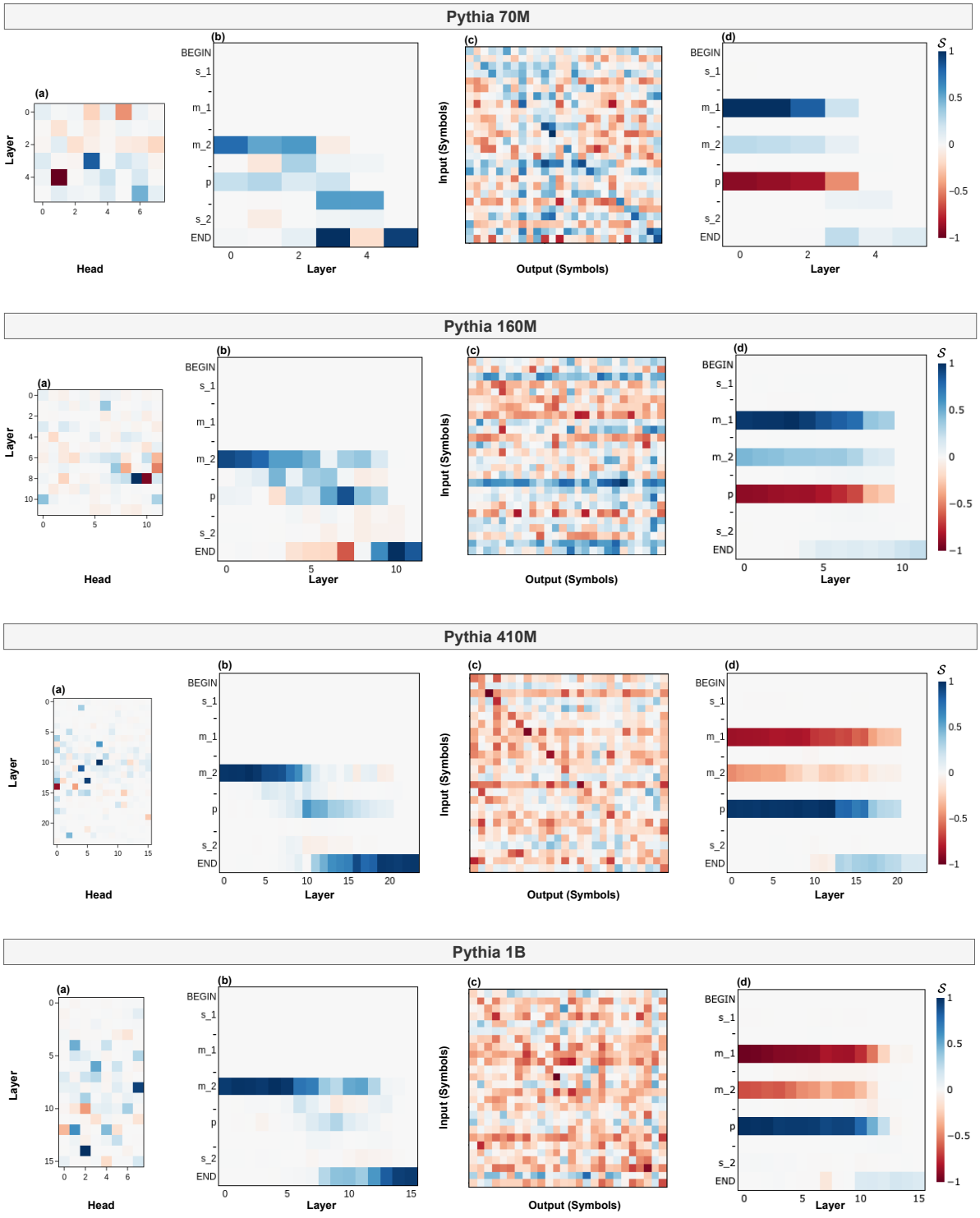

Figure 11: Comprehensive results of the symbolic circuit analysis across different series of models (Pythia). (a) Attention output patching results and (b) residual stream patching results in the middle-term intervention setup. (c) OV circuit logit lens results for m-suppression head, with input and output comprising 26 uppercase letters. (d) Residual stream patching results in the all-term corruption setup. For clarity, a dash (–) indicates the averaged values for tokens appearing between terms.

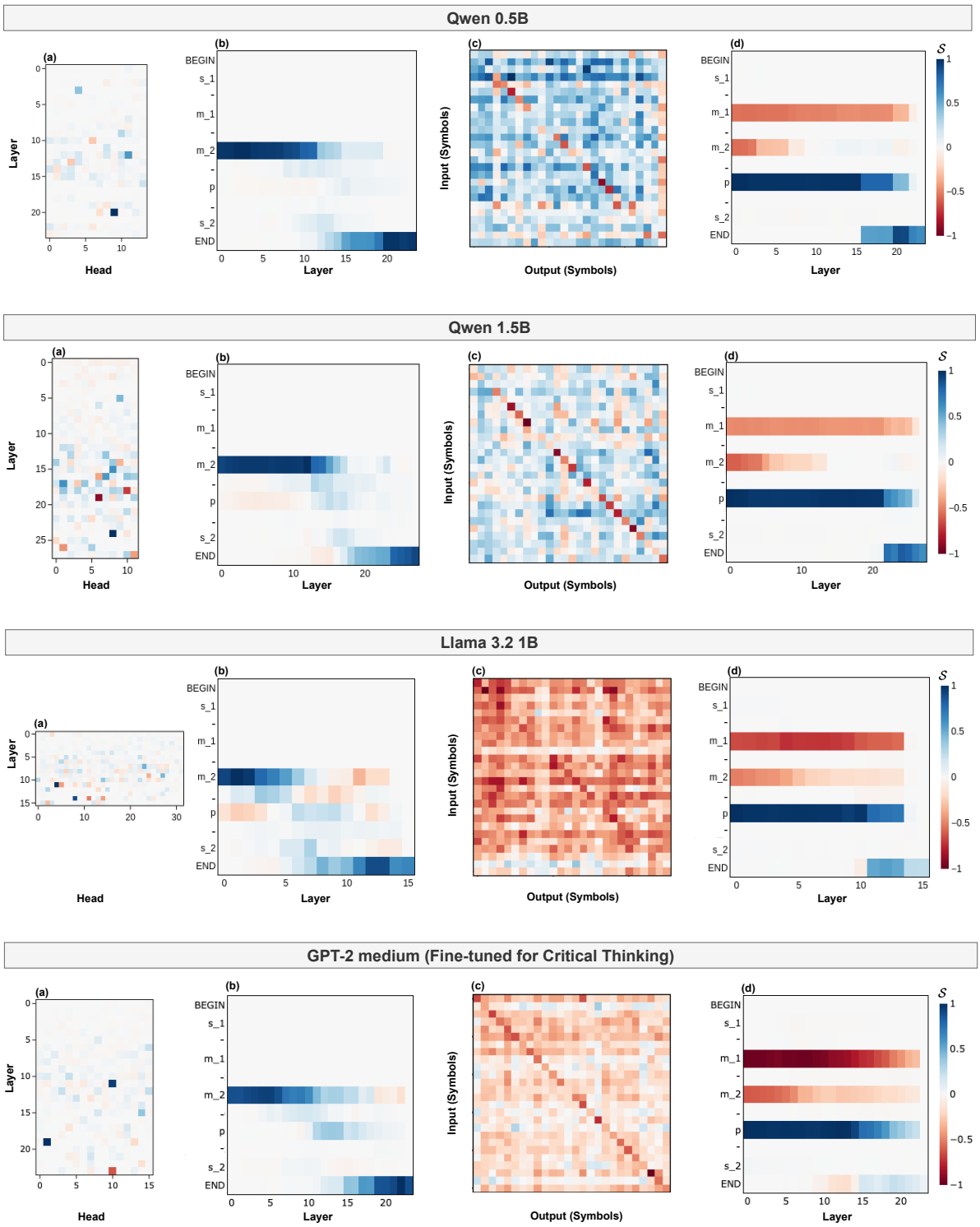

Figure 12: Comprehensive results of the symbolic circuit analysis across different series of models (Qwen2.5, Llama3.2 and fine-tuned GPT-2 medium). (a) Attention output patching results and (b) residual stream patching results in the middle-term intervention setup. (c) OV circuit logit lens results for m-suppression head, with input and output comprising 26 uppercase letters. (d) Residual stream patching results in the all-term corruption setup. For clarity, a dash (–) indicates the averaged values for tokens appearing between terms.