Emergence of Minimal Circuits for Indirect Object Identification in Attention-Only Transformers

Rabin Adhikari
Saarland University
66123 Saarbrücken
raad00002@stud.uni-saarland.de

Abstract

Mechanistic interpretability aims to reverse-engineer large language models (LLMs) into human-understandable computational circuits. However, the complexity of pretrained models often obscures the minimal mechanisms required for specific reasoning tasks. In this work, we train small, attention-only transformers from scratch on a symbolic version of the Indirect Object Identification (IOI) task, a benchmark for studying coreference-like reasoning in transformers. Surprisingly, a single-layer model with only two attention heads achieves perfect IOI accuracy, despite lacking MLPs and normalization layers. Through residual stream decomposition, spectral analysis, and embedding interventions, we find that the two heads specialize into additive and contrastive subcircuits that jointly implement IOI resolution. Furthermore, we show that a two-layer, one-head model composes information from the previous layer primarily through query-key interactions. These results demonstrate that task-specific training induces highly interpretable, minimal circuits, offering a controlled testbed for probing the computational foundations of transformer reasoning.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success across a vast range of natural language processing tasks (DeepSeek-AI, 2025; Liu et al., 2026; OpenAI, 2026; Team, 2025, 2024; Yang et al., 2025). Yet, their internal operations remain largely opaque, motivating the field of mechanistic interpretability, which seeks to reverse-engineer these "black boxes" into understandable circuits and algorithms (Cammarata et al., 2020; Nanda et al., 2023; Olah et al., 2020). Its ultimate goal is to achieve a circuit-level understanding where individual components like neurons and attention heads are mapped to specific algorithmic roles (Conmy et al., 2023; Elhage et al., 2021). However, the immense scale, residual connections, and non-linearities of modern LLMs present significant challenges to this endeavor.

To navigate this complexity, researchers often start with simplified or "toy" models as controlled environments for developing and validating interpretability tools (Chughtai et al., 2023; Elhage et al., 2022; Furuta et al., 2024; Geva et al., 2021; Heimersheim and Janiak, 2023; Nanda et al., 2023). By training models on constrained, synthetic objectives, we can reduce confounding variables from complex linguistic structures and discover core computational mechanisms in a cleaner setting.

A common approach to understanding these models involves analyzing pre-trained transformers (Vaswani et al., 2017) on specific tasks they can perform (Bereska and Gavves, 2024; Brinkmann et al., 2024). To investigate these capabilities, we focus on the Indirect Object Identification (IOI) task. Wang et al. (2023) showed that GPT-2 small (Radford et al., 2019) implements IOI through a multi-hop attention circuit involving distinct classes of heads. However, this mechanism arises within a model pretrained for next-token prediction on natural text, which is an inherently complex optimization setting.

In contrast, we train minimal, attention-only transformer models (Vaswani et al., 2017) from scratch exclusively on a symbolic version of the IOI task. We find that a straightforward model, a single-layer transformer block with just two attention heads, can solve this task perfectly. Because the IOI task requires dynamic duplicate-token detection and exclusionary copying, the computations proven to be beyond the representational capacity of bigram and skip-gram models (Elhage et al., 2021; Wang et al., 2023), our findings build on Shlegeris (2023) by demonstrating exactly how a minimal model with a single attention layer implements this logic. Furthermore, a detailed analysis of this model uncovers a highly compact and in-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 36–46
July 2-7, 2026 ©2026 Association for Computational Linguistics

terpretable circuit where the solution is computed via a direct additive combination of the two heads' outputs, rather than a complex, multi-hop pipeline found in GPT-2 small (Radford et al., 2019).

Our contributions are threefold:

1. We demonstrate that a one-layer, two-head attention-only model is sufficient to solve the IOI task with a fixed template perfectly.

2. We provide a mechanistic analysis that uncovers a minimal circuit based on an additive combination of specialized attention head outputs.

3. We argue that the circuits in large, broadly pre-trained models may be overly complex due to multi-task pressures, whereas task-constrained training can reveal more parsimonious mechanisms.

2 Background

2.1 Task: Indirect Object Identification (IOI)

The IOI task serves as a standardized benchmark for studying coreference-like reasoning and dynamic memory mechanisms within language models (Ensign and Garriga-Alonso, 2024; Wang et al., 2023). In a typical natural language IOI sentence, an initial dependent clause introduces two distinct names: a subject (S) and an indirect object (IO). Subsequently, the main clause repeats the subject. The model's objective is to accurately predict the IO as the next logical token. For example, in the sentence "When John and Mary went to the store, John gave a drink to ___," the model must predict "Mary". Following Wang et al. (2023), which identified a complex, multi-hop circuit for this task in GPT-2 small (Radford et al., 2019), our work investigates this fundamental exclusionary logic in a drastically simplified setting.

2.2 Transformer Architecture

To establish our mathematical notation, we rely on the framework introduced by Elhage et al. (2021) for reverse-engineering attention-only transformers. This framework conceptualizes the transformer's residual stream as a primary communication channel where different components read and write information independently.

2.2.1 Residual Stream Decomposition

For an attention-only model, the state of the residual stream at layer l, denoted as $x^{(l)}$, is strictly a linear combination of the initial embeddings and the outputs of all preceding attention heads. Let x_{embed} and x_{pos} denote the token and positional embeddings, respectively. The residual stream just before unembedding in a single-layer model is formalized as follows.

$$x^{(1)} = x_{embed} + x_{pos} + \sum_{h=1}^{H} \text{out}_h^{(0)}$$

where $\text{out}_h^{(0)}$ is the output vector written to the residual stream by head h in layer 0. The final logit prediction for any token t is computed by projecting this residual stream onto the unembedding matrix W_U: $L_t = (x^{(1)})^T W_{U[:,t]}$.

2.2.2 Transformer Circuits

To compute its output, each attention head h reads from the residual stream using three projection matrices: the Query matrix (W_Q^h), the Key matrix (W_K^h), and the Value matrix (W_V^h). The queries and keys interact to determine the attention scores between tokens, while the values determine the information moved across the sequence, which is then projected back into the residual stream via an output matrix (W_O^h).

We can analyze the behavior of individual heads by decomposing these operations into two distinct circuits (Elhage et al., 2021). The QK (Query-Key) circuit dictates the attention scores between a query token and a key token, represented by the end-to-end transition matrix $M_{QK}^h = W_E^T (W_Q^h)^T W_K^h W_E$, where W_E denotes the token embedding matrix. The OV (Output-Value) circuit dictates how attending to a specific token updates the final output logits, represented by the transition matrix $M_{OV}^h = W_U W_O^h W_V^h W_E$. Because these exact matrices, M_{QK}^h and M_{OV}^h, map directly from the vocabulary space back to the vocabulary space, they serve as the foundation for our spectral analysis in section 4.2.3.

2.2.3 Composition of Attention Heads

In multi-layer attention-only architectures, transformer heads develop functional hierarchies by composing information across layers (Elhage et al., 2021). Because the input to a head in a subsequent layer j is the residual stream $x^{(j)}$, which contains the outputs of heads from an earlier layer i ($i < j$), the projection matrices of layer j directly read the information written by layer i. This interaction is formally categorized into three types of composition described below.

Q-Composition The output of Layer i is projected through the Query matrix of Layer j, affecting what the latter head attends to.

K-Composition The output of Layer i is projected through the Key matrix of Layer j, altering how the latter head matches incoming queries.

V-Composition The output of Layer i is projected through the Value matrix of Layer j. This modifies the actual information the later head moves across the sequence, effectively creating a "virtual attention head."

While both Q- and K-composition enable more complex attention routing by acting on different sides of the attention score calculation, V-composition strictly affects information transfer. We leverage this framework in section 4.3.3 to perform targeted ablations, identifying precisely which pathways our two-layer model relies upon to solve the IOI task.

3 Dataset and Model Configuration

3.1 The IOI Task in a Symbolic Setting

To isolate the core relational reasoning challenge of the IOI task described above, we construct a purely symbolic dataset. This formulation abstracts away all linguistic and tokenization complexities, enabling precise inspection of what the model must represent to distinguish between "subject" and "object" roles without natural language confounds.

The training data consists of 6-token sequences following the format `<BOS> IO S1 S2 <MID> ?`. The `IO`, `S1`, and `S2` are two unique names drawn from a small vocabulary. The model's task is to predict the name token that is not repeated before the `<MID>` token.

Essentially, the dataset follows two rigid templates, depending on the order of the names. Using "John" and "Mary" as examples, the templates are:

1. `<BOS> John Mary Mary <MID> John`
2. `<BOS> John Mary John <MID> Mary`

Following Wang et al. (2023), we refer to the first template as "BABA" and the second as "BAAB".

3.2 Model Configuration

To maximize interpretability, we used simple attention-only transformer models with absolute positional embeddings. Feed-forward networks and layer normalization were omitted to isolate the function of the attention mechanism. The vocabulary consists of 6 name tokens plus the two special tokens `<BOS>` and `<MID>`, for a total size of 8. The residual stream dimension was kept the same size as the vocabulary (8), and the head dimension was $d_{head} = {}^{8}/_{N_h}$, where N_h is the number of heads. In this formulation, the number of parameters in the model is independent of the number of heads.

3.2.1 Training and Analysis Setup

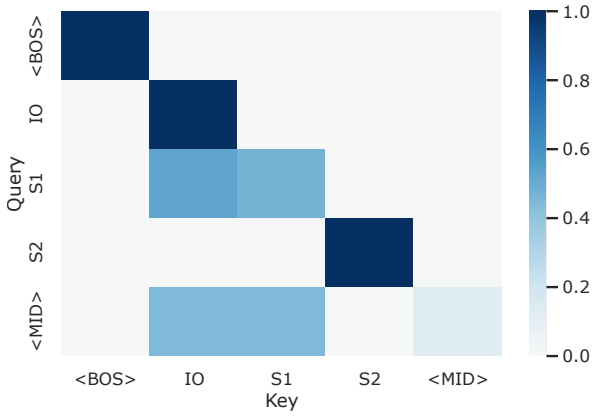
Models were trained from scratch on the symbolic IOI dataset using a cross-entropy loss to predict the token at the `<MID>` position. Each training batch contained all 60 possible unique sequences ($6 \times 5 = 30$ ways of picking the names in the dependent clause and two ways of ordering them in the main clause). We used the AdamW (Loshchilov and Hutter, 2019) optimizer with the OneCycle (Smith and Topin, 2019) learning rate scheduler, with a maximum learning rate of 0.1 and weight decay of 0.01. Training and analyses were performed on a single NVIDIA A40 GPU using PyTorch (Paszke et al., 2019) and TransformerLens (Nanda and Bloom, 2022) libraries.

4 Results and Analysis

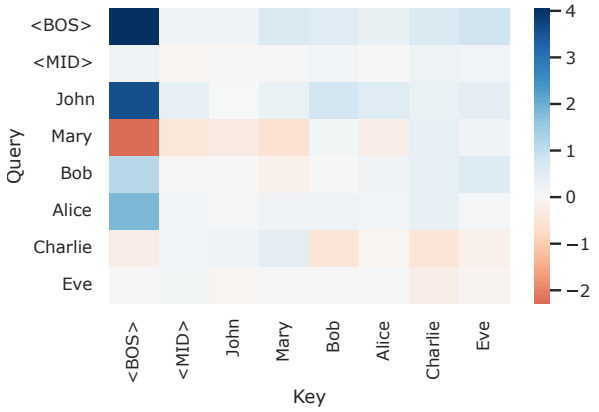
4.1 Zero-Layer and Single-Head Baselines

A zero-layer model (acting as a bigram) predicts every name token with an equal probability of $\approx {}^{1}/_{N_{names}} = 16.7\%$, as the `<MID>` token must predict a name without utilizing prior context. When extending to a single-layer, single-head model, it assigns ≈ 0.5 probability for the names provided in the prompt. However, it cannot distinguish which one is correct. As shown in figure 1a, the `<MID>` token attends roughly equally to both the names in the dependent clause, indicating that a single attention head cannot jointly encode the information required to **(i)** identify which token serves as the correct referent and **(ii)** propagate that information to the prediction position. The roles of "reference detection" and "copying" appear to be functionally incompatible within the attention mechanism with a single head.

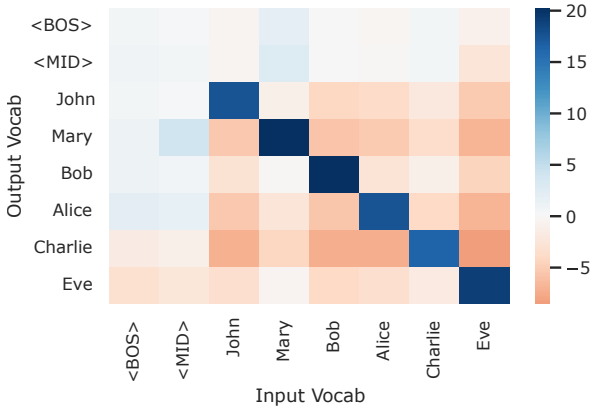
We analyze the QK and OV circuits to understand the failure mode of the single-head model. From the QK circuit (see figure 1b), we observe that the `<MID>` token attends almost uniformly to all tokens. And the OV circuit (see figure 1c) shows that each name token makes a high positive contribution to its own logit but small negative contri-



(a) Average Attention Heatmap



(b) The QK Circuit



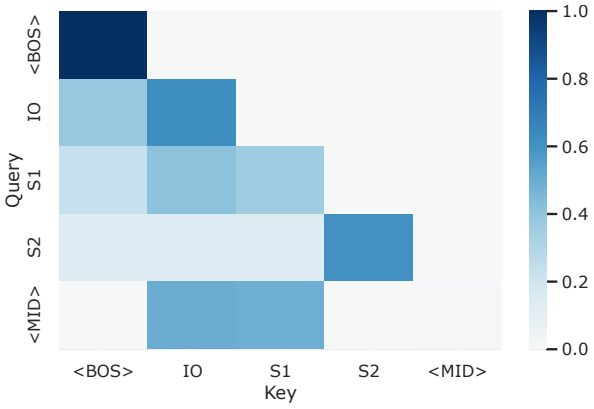
(c) The OV Circuit

Figure 1: Single-head, one-layer models fail at the IOI task. The attention and circuit analysis reveal that a single head cannot simultaneously detect the correct reference and copy it to the output.

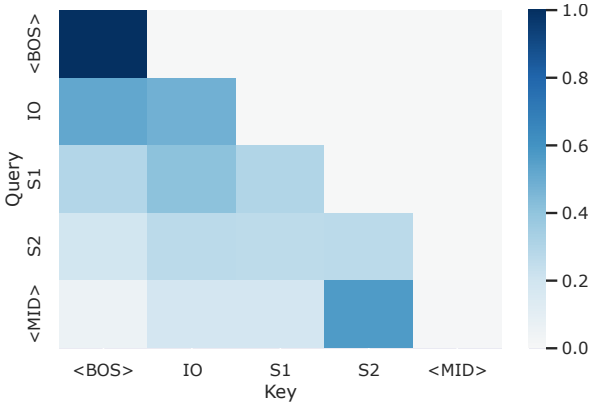
butions to the logits for other names. The uniform attention pattern averages these contributions, resulting in similar logits for both names.

4.2 A Two-Heads, One-Layer Model Learns IOI Perfectly

When the model with a single attention layer is extended to two attention heads, it achieves perfect accuracy on the IOI task. Figure 2 shows distinct



(a) The first head attends equally to both context names.



(b) The second head distributes attention between the main clause subject and the names in the dependent clause.

Figure 2: Average attention heatmap for a two-heads, one-layer model.

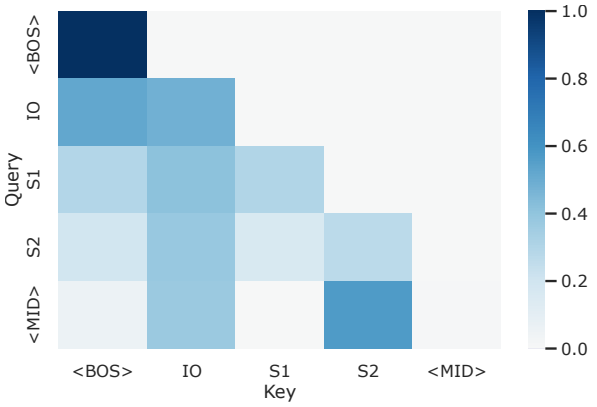
attention patterns of the two heads, averaged across all the possible inputs.

4.2.1 Two Heads with Distinct Roles

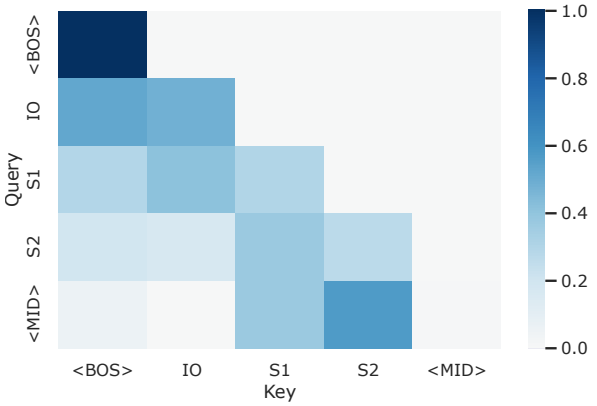
We observed that, for both the templates of our symbolic dataset, the first head consistently attends to the two names in the initial dependent clause (see figure 2), indicating its role in identifying the relevant referents. The second head, however, always attends to the subject of the main clause and the non-repeated name in the dependent clause (see figure 3). So, this head does most of the heavy lifting, finding out the unique set of tokens to attend to. Furthermore, it suggests that the second head is responsible for integrating the referential information with the context provided by the main clause to determine the correct output.

4.2.2 Residual Stream Decomposition

To understand how the model's components contribute to the final prediction, we decompose the residual stream at the final token position (corresponding to <MID> token) into the contributions



(a) The second head attends to `IO` and `S2` for the "BAAB" template.



(b) The second head attends to `S1` and `S2` for the "BABA" template

Figure 3: The attention map for the second head depends on the template. It dynamically attends to the subject of the main clause and the non-repeated name of the dependent clause.

of those components. We then project these components onto directions in the embedding space corresponding to the correct and incorrect names, as well as their sum and difference, using LogitLens (nostalgebraist, 2020). Figure 4 shows that the first head's output is aligned closest with the sum direction, i.e., it represents the combined contribution of both the correct and incorrect names (*correct* + *incorrect*). On the other hand, the second head's output aligns closest with the direction of the token difference, i.e., the contrast between the two name embeddings (*correct* – *incorrect*). Since the final logits are computed by adding the contribution of all the components, the logit component for the incorrect token roughly cancels out, and the direction corresponding to the correct token is amplified.

This analysis is also not foolproof because we can see that the second head also has some component in the direction of the correct token, as well

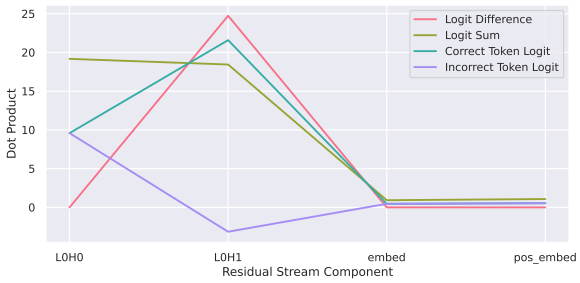


Figure 4: Residual stream decomposition for two-head, one-layer model. Projections of the residual stream components onto the correct and incorrect directions, along with their sum and difference.

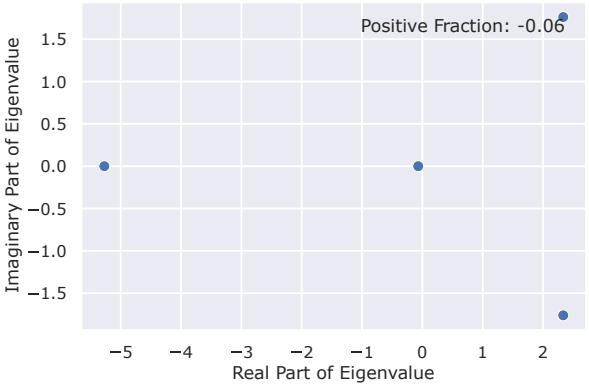
as the sum direction. Nevertheless, we observe a clear division of labor between the two heads: one aggregates signals (additive), while the other suppresses the incorrect alternative (contrastive). Together, they form an additive-contrastive circuit to produce a clean, interpretable mechanism for generating the correct logits.

4.2.3 Spectral Analysis of QK and OV Circuits

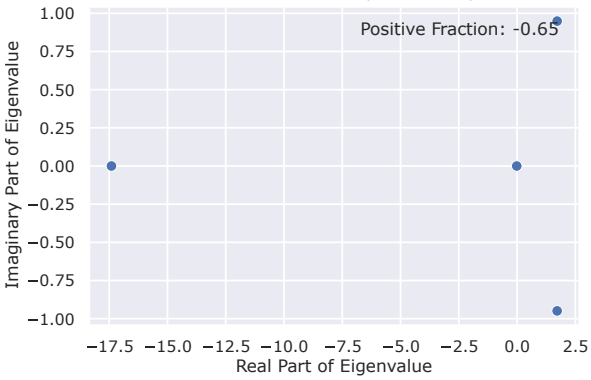
While random matrices typically exhibit symmetric eigenvalue distributions (Tarnowski, 2022), the QK and OV matrices in our two-head model display significant asymmetry, reinforcing their specialized functional roles (see figures 5 and 6). Furthermore, on the top-right of each subfigures, we report the fraction of positive eigenvalues for each head calculated using the formula $\frac{\sum_i \lambda_i}{\sum_i |\lambda_i|}$, where λ_i are the eigenvalues of the matrix and $|\lambda_i|$ are their magnitudes.

Spectral Properties of QK Circuits Observing the eigendecomposition of the QK matrices in figure 5, we notice that the first head has a moderate suppression mechanism, denoted by a real eigenvalue of ≈ -5.2, to forbid attending to some dimensions (or some tokens), indicating a less pronounced inhibitory effect. Additionally, it has two other complex eigenvalues $\approx 2.3 \pm 1.6i$, indicating some amplifying effect along with some rotation in some dimension. Finally, the positive fraction around zero (-0.06) suggests that the amplifying effect of the rotational component is almost balanced by the suppressive effect of the negative eigenvalue. Hence, the overall attention dynamics of the first head are relatively neutral.

The second head has a strong suppression mechanism (≈ -17.5), indicating a more pronounced

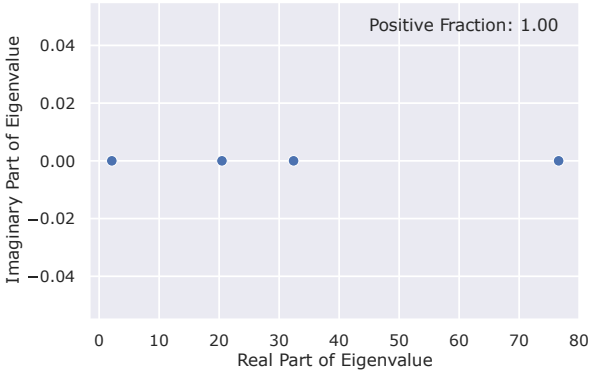


(a) The first head exhibits relatively neutral dynamics.

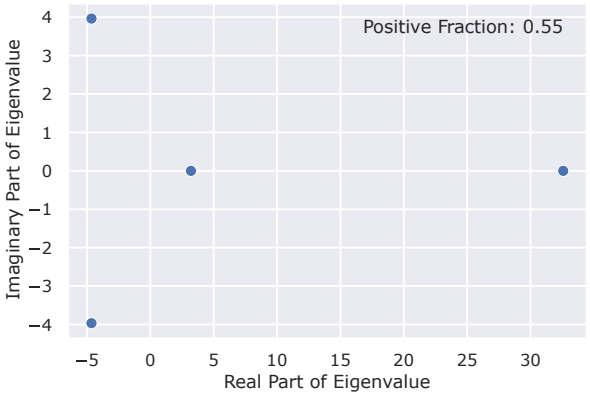


(b) The second head indicates a strong suppressive effect.

Figure 5: Eigenvalue distribution of QK circuits for a two-heads, one-layer model.



(a) The first head acts purely additively.



(b) The second head suggests a mix of additive and contrastive (rotational) contributions.

Figure 6: Eigenvalue distribution of OV circuits for a two-heads, one-layer model.

inhibitory effect. The positive fraction of -0.65 suggests that the suppressive effect of the dominant negative eigenvalue outweighs the amplifying effect of the rotational components (denoted by complex eigenvalues), leading to an overall inhibitory attention dynamics.

Spectral Properties of OV Circuits The eigendecomposition of the OV matrices further reveals the asymmetry between the two heads (see figure 6). The first head is a **copying** or **passthrough** head, which identifies important tokens via its QK circuit and then amplifies their presence in the residual stream without any rotation or inversion.

The second head has half of its eigenvalues as real and positive, while the other half are imaginary with negative real parts. This suggests that the chosen token will copy itself in some dimension and rotate with inversion in another dimension, which can be interpreted as subtracting from the logits of the other token with some added transformations.

This distinction corresponds naturally with the roles inferred from embedding projections: one head aggregates signals (additive), while the other suppresses the incorrect alternative (contrastive).

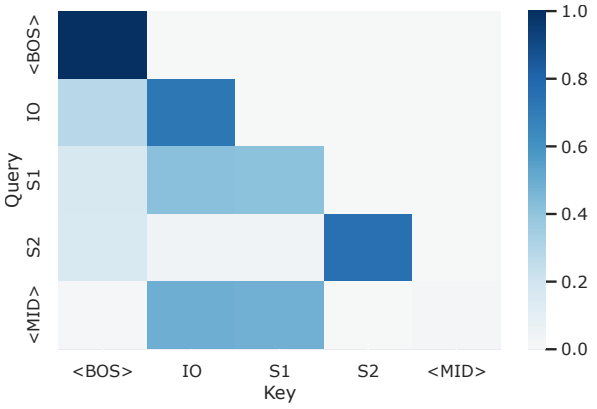
4.2.4 Positional Focus of Attention Heads

To isolate the model's reliance on positional embeddings, we assign a single, averaged embedding to all name tokens, removing the model's ability to differentiate them by identity. From figure 7, we can consider the first head as a **positional head** that focuses on the positions of the names in the dependent clause, independent of their word embeddings, because the attention patterns in figures 2a and 7a look the same.

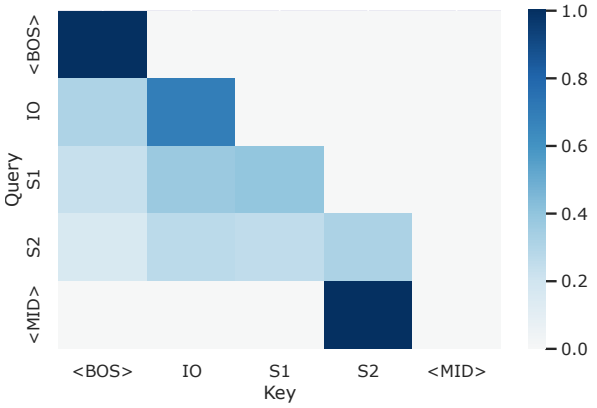
The second head attends predominantly to the position of occurrence of the subject in the main clause. However, despite this positional focus, it attends to the name in the dependent cause, not repeated in the main clause. This indicates that the second head is responsible for integrating positional as well as contextual information to determine the correct output.

4.2.5 Ablation: Positional Embeddings

To study how the model utilizes positional information, we train the same model architecture without any positional embeddings. The model achieves an accuracy of $\approx 70\%$ on the IOI task, with $\approx 67\%$

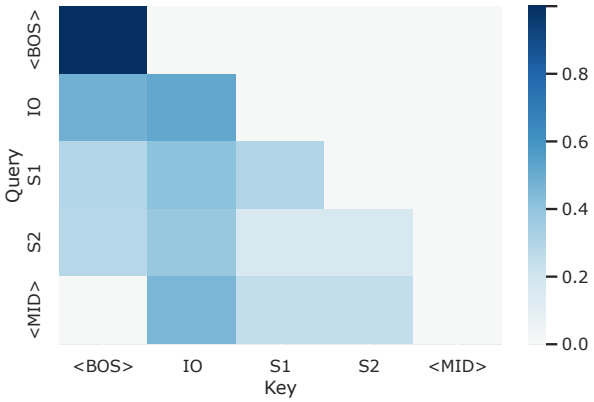


(a) The first head focuses on the positional embeddings of the names in the dependent clause.

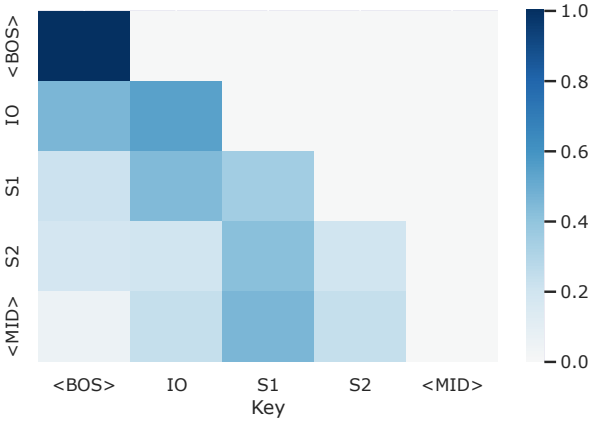


(b) The second head attends primarily to the positional embedding of the subject of the main clause.

Figure 7: Average attention heatmap for two-head, one-layer model with name embeddings averaged.



(a) The head focuses only on the name tokens and most on IO for the "BAAB" template.



(b) The head focuses only on the name tokens and most on S1 for the "BABA" template.

Figure 8: Average attention heatmap for the first head in a two-head, one-layer model trained without positional embeddings. Both heads focus on the name tokens of the prompt and focus most on the correct output name present in the dependent clause.

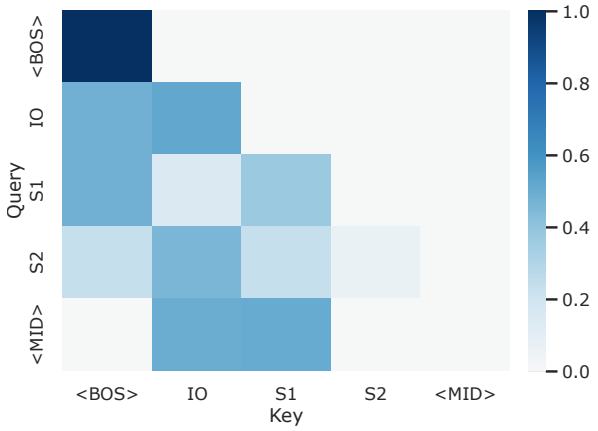
probability on the correct token, indicating that positional embeddings are not strictly necessary for the model to learn the task. Unlike the ones with positional embeddings, the heads trained in this manner exhibit similar attention patterns, managing to focus primarily on the correct non-repeated name (see figure 8). This suggests the model can fall back on purely semantic contextual relationships, though explicit positional cues drastically simplify the optimization landscape, allowing it to reach 100% accuracy and perfectly decouple into additive-contrastive roles.

4.3 A Two-Layers, One-Head Model

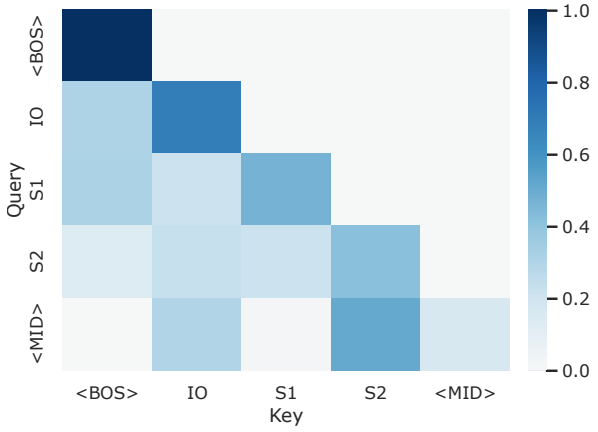
We also train a two-layer attention-only transformer with one head in each layer to observe how a model performs IOI in the availability of compositions (Elhage et al., 2021). The head dimension was 4 for the one-layer two-heads model. For this model, since we have a single head per layer, the head dimension is the same as the hidden dimension, i.e., 8. So, this model has more representational capacity than the one-head one-layer model. If this model doesn't perform any composition, i.e., if the second layer just doesn't depend on the output of the first layer, then it is the same as the one-layer two-heads model with each layer writing to the residual stream in orthogonal subspaces of 4 dimensions each.

4.3.1 Attention Heatmap

Figures 9 and 10 show the attention heatmap for a two-layer, one-head model, averaged across "BAAB" and "BABA" templates, respectively. We observe that the attention patterns of both layers change depending on the template. From figures 9a and 10a, we observe that the `<MID>` token in the first layer still attends to both the name tokens in the dependent clause for both templates, similar to the first head of the one-layer two-heads model. However, the `S2` token in the first layer changes

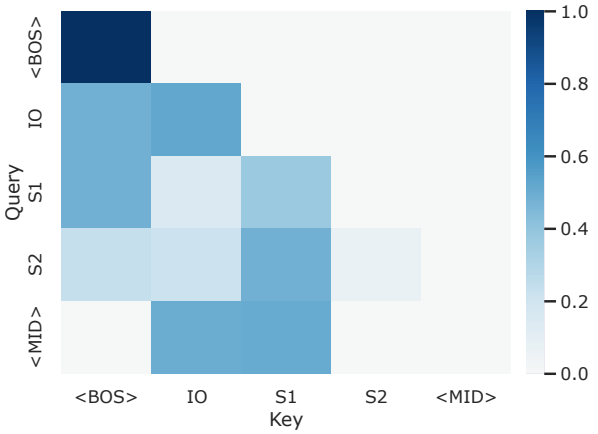


(a) Attention heatmap for the first layer's head.

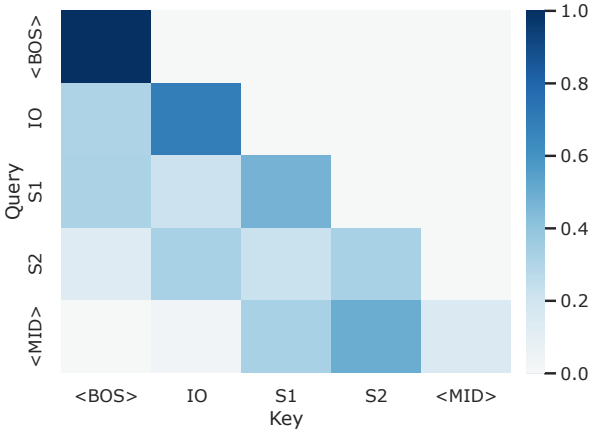


(b) Attention heatmap for the second layer's head.

Figure 9: Attention heatmap for a two-layer, one-head model for the "BAAB" template.



(a) Attention heatmap for the first layer's head.



(b) Attention heatmap for the second layer's head.

Figure 10: Attention heatmap for a two-layer, one-head model for the "BABA" template.

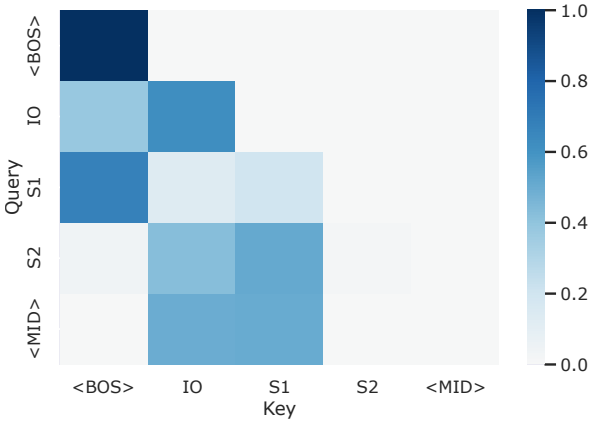
its attention pattern based on the template; it attends more to the IO token than the S1 token for the "BAAB" template and more to the S1 than the IO token. So, the first head is not solely positional, but aggregates information to S2 token to be used by the latter head. Although the attention pattern of the second layer for the <MID> token (see figures 9b and 10b) seems almost similar to the attention pattern of the second head of the one-layer two-heads model (see figure 3), this time it attends to the aggregated information from the first layer.

4.3.2 Role of Positional Embeddings

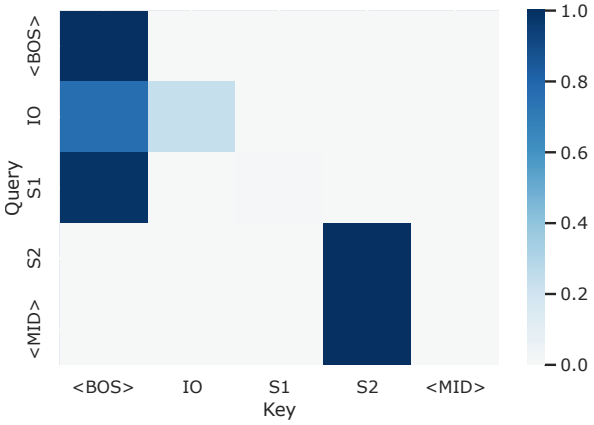
Similar to section 4.2.4, we analyze the attention pattern of the heads when the distinction among the names is removed (see figure 11). We observe that the attention pattern in figure 11a is very similar to figure 7a, indicating that it has a strong positional focus. However, although the <MID> token in the second layer (see figure 11b) attends primarily to the token before it, it changes its attention pattern based on the context provided by the first layer. So, the second head is not solely positional, but integrates positional as well as contextual information to determine the correct output.

4.3.3 Ablation: Q, K, and V-Composition

To study the type of composition that the model is performing (Q, K, or V), we ablate them one by one by subtracting the output of the first layer from the corresponding input of the Q, K, and V matrices. We observe a drop in accuracy in the following order: Q-composition ($\approx 100\%$ drop), V-composition ($\approx 93.33\%$ drop), and K-composition ($\approx 26.67\%$ drop). This indicates that the model is heavily relying on the Q and V-compositions to perform the task. So, we can conclude that the two-layer one-head model is indeed performing some composition to solve the IOI task, different from the one-layer two-heads model. This hints that finding a circuit capable of solving a given task using composition is an easier task for the optimizer than building two orthogonal subspaces in the residual stream.



(a) The head on the first layer is a positional head with almost the same attention pattern with indentifiable name embeddings.



(b) The head on the second layer pays strong attention to the positional embedding of the subject of the main clause.

Figure 11: Attention heatmap for two-layer, one-head model with averaged name embeddings. We see a strong positional focus in both heads.

5 Conclusion

In this study, we showed that a single-head single-layer attention-only transformer can't solve a symbolic version of the Indirect Object Identification (IOI) task. However, if we increase the number of attention heads to two, keeping the number of parameters the same, it can perfectly solve it. Our mechanistic analysis revealed an elegant division of labor: one head aggregates referential information additively, while the other performs contrastive suppression of incorrect alternatives. In a two-layer, single-head model, we further observed compositional behavior across layers, indicating the emergence of functional hierarchy. These findings highlight that task-constrained training can produce parsimonious and interpretable circuits, offering valuable insight into the primitive computational motifs that may underlie reasoning in larger, pretrained language models.

Limitations

While this work successfully isolates minimal computational motifs for coreference-like reasoning, our analysis is bounded by the following constraints.

Sensitivity to Sequence Structure By abstracting away linguistic complexity into rigid 6-token sequences, we successfully isolated the core exclusionary logic of IOI. However, this paper doesn't explore how this minimal circuit behaves when subjected to varying sequence lengths, multiple interdependent clauses, or dynamic syntax, and at what threshold of structural complexity this two-head circuit necessitates the multi-hop mechanisms described by Wang et al. (2023).

Training Dynamics Our mechanistic analysis focuses exclusively on the fully converged model. We do not investigate the developmental interpretability or training dynamics that lead to the emergence of these specialized circuits. Specifically, it is currently unknown at what phase during the optimization process the two heads differentiate into their respective additive and contrastive roles, or what specific loss landscape dynamics drive this strict division of labor.

References

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

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

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. 2020. Thread: Circuits. *Distill*, 5(3):e24.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. 2023. A toy model of universality: Reverse engineering how networks learn group operations. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 6243–6267. PMLR.

Arthur Conmy, Augustine N. 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 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.

DeepSeek-AI. 2025. DeepSeek-V3.2: Pushing the frontier of open large language models. *CoRR*, abs/2512.02556.

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

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.

Danielle Ensign and Adrià Garriga-Alonso. 2024. Investigating the indirect object identification circuit in mamba. In *ICML 2024 Workshop on Mechanistic Interpretability*, Vienna, Austria. OpenReview.net.

Hiroki Furuta, Gouki Minegishi, Yusuke Iwasawa, and Yutaka Matsuo. 2024. Towards empirical interpretation of internal circuits and properties in grokked transformers on modular polynomials. *Transactions on Machine Learning Research*, 2024.

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

Stefan Heimersheim and Jett Janiak. 2023. A circuit for python docstrings in a 4-layer attention-only transformer. AI Alignment Forum.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. Ministral 3. *CoRR*, abs/2601.08584.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Neel Nanda and Joseph Bloom. 2022. TransformerLens. V2.16.1.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

nostalgebraist. 2020. Interpreting GPT: the logit lens. LessWrong.

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.

OpenAI. 2026. OpenAI GPT-5 system card. *CoRR*, abs/2601.03267.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Buck Shlegeris. 2023. One-layer transformers aren't equivalent to a set of skip-trigrams. AI Alignment Forum.

Leslie N. Smith and Nicholay Topin. 2019. Super-convergence: Very fast training of residual networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, SPIE.

Wojciech Tarnowski. 2022. Real spectra of large real asymmetric random matrices. *Physical Review E*, 105(1):L012104.

Gemma Team. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Llama Team. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.