How Language Models Combine Multiple Binding Mechanisms

Anonymous authors
Paper under double-blind review

Abstract

Language models must bind entities together in context to later retrieve relevant information, for example when remembering which city a person lives in. Prior work has established three mechanisms (positional, lexical, and reflexive) that models use for this retrieval, but left open where the underlying binding signals are stored and how they are combined into an answer. In this work, we isolate the low-rank subspaces of the final token residual stream used by each mechanism and show that they can be individually manipulated. Further, we show that the subspaces contribute additively to the query key product of a primary attention head driving answer retrieval. Interventions on individual subspaces selectively redirect the head's attention and the model's final output. Finally, we examine the subspaces themselves: the positional subspace primarily encodes fact order, while the lexical and reflexive mechanisms share a semantic identity channel. Taken together, these findings show how known retrieval heuristics are represented and combined. More broadly, the work provides insight into how redundant mechanisms are implemented and reconciled within language models.

1 Introduction

When a language model processes a passage, it needs to keep track of which entities go together so that it can use that information later, for example remembering that Alice lives in Paris. This capacity underlies in-context reasoning generally, and its analog in human cognition, the binding problem, has been studied for decades (Fodor & Pylyshyn, 1988; Smolensky, 1990; Treisman, 1996; Greff et al., 2020). Large language models now reliably demonstrate this capability, and recent interpretability work has begun to explain how.

Feng & Steinhardt (2024) found that models tag entities and attributes with additive binding ID vectors; subsequent work showed that these IDs are, in the tasks studied, largely *order* IDs. Dai et al. (2024) localized them to a low-rank subspace graded with list position, and Prakash et al. (2024; 2025) traced circuits that transmit and dereference positional pointers. However, position is not the whole story. Gur-Arieh et al. (2025) showed that with more facts in context (e.g. lists of twenty facts), models retrieve using three mechanisms: the positional mechanism keyed to the fact's index in the list, a lexical mechanism keyed to the identity of the queried entity, and a reflexive mechanism keyed to a pointer to the answer entity already resolved by an earlier layer. Most recently, Lee et al. (2026) associated query-key subspaces with the positional and lexical binding mechanisms in a retrieval task.

In this work, we ask where the retrieval signals live in the residual stream, what they represent, and how the model combines them into a final answer.[1] We isolate subspaces of the residual stream for each mechanism and show that they can be manipulated independently. We then follow these components through the query of the primary retrieval head at the final token position and show that they contribute additively to the attention logits, with softmax collapsing the votes from the different subspaces and driving the attention to the final answer for retrieval. We also examine the subspaces themselves. We find that while the positional subspace encodes fact order as expected, the lexical and reflexive subspaces seem to share a content-matching channel, matching on a representation of the entity itself, rather than a tag or hash for example.

[1] Code: `https://anonymous.4open.science/r/binding-mechanisms`

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

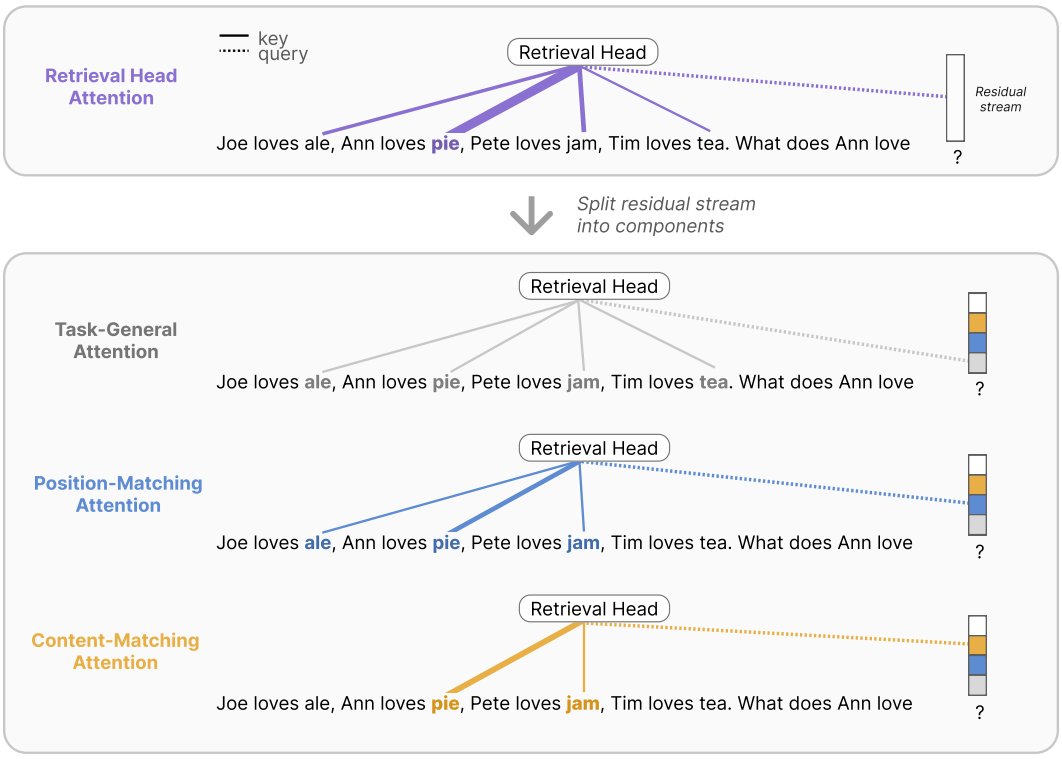


Figure 1: **Attention of the primary retrieval head.** We decompose the residual stream into a generic task component and mechanism subspaces, and these subspaces drive the attention of the primary retrieval head to their targets.

Beyond retrieval tasks, our results characterize one way a language model handles redundant mechanisms. Each mechanism writes its vote into the residual stream and selection occurs in the attention computation of a specific head, where softmax decides on a winner. The fact that we are able to causally manipulate individual votes even when they do not determine the final answer provides insight into how a model that appears to have decisively chosen a strategy may in fact have considered multiple strategies in parallel up until the point of retrieval. Whether other examples of redundant mechanisms resolve in the same way is an open question.

Overall, our contributions are as follows:

- We isolate and individually manipulate low rank residual stream subspaces for the three retrieval mechanisms.
- We show that the subspaces combine additively in the query of a primary retrieval head that reads all three signals at once.
- We identify a shared content-matching overlap in the lexical and reflexive subspaces.

2 Experimental Setup

We report our primary findings on Gemma 2-2B (Gemma Team et al. (2024)) and replicate on Llama 3.1-8B (Grattafiori et al. (2024)). We use the boxes task (Kim & Schuster (2023), Prakash et al. (2024)) with 20 facts, with objects pulled from a pool of 82 household items and box labels from single letters, in line with Gur-Arieh et al. (2025). Each example consists of 20 facts of the form "The apple is in Box C", followed by either a box $\rightarrow$ object query ("What does Box C contain?") or an object $\rightarrow$ box query ("Which box is the apple in?"). Gemma 2-2B answers correctly on 99.2% of box $\rightarrow$ object prompts and 98.8% of object $\rightarrow$ box prompts (Llama 3.1-8B: 99.4% and 99.8%, Appendix A).

We also utilize Gur-Arieh et al. (2025)'s counterfactual prompt method where for each base prompt, a counterfactual prompt is created such that each of the three mechanisms (positional, lexical, reflexive) predicts a different answer. This is accomplished by using the same 20 objects and box letters but rebinding them into new facts and reordering them. This makes it such that when the counterfactual is patched into a base prompt, the positional mechanism now predicts the token at the counterfactual prompt's answer fact index, the lexical mechanism the entity bound to the new queried entity, and the reflexive mechanism the counterfactual prompt's answer token. Counterfactual patching is performed by patching the full residual stream of the counterfactual prompt into a base prompt at one to five final token positions. Which content mechanism wins depends on word order (Appendix B). Pooled across query formats, the reflexive mechanism wins 53.5% of counterfactual prompts when the answer comes before the queried entity in its fact, versus 16.0% when it comes after. This is reverse for the lexical mechanism (1.8% vs. 32.2%). We study each mechanism in the setting it's dominant in. Several experiments require various modifications to the setup and are explained in their respective sections.

3 Isolating the Retrieval Heads

In order to find out how the model decomposes the query-key interactions of the different mechanisms, we first identify which attention heads execute the retrieval task and whether this is split by mechanism or shared. We use Direct Logit Attribution (DLA) (Elhage et al., 2021) to identify candidates, then examine their causal role with targeted patching. L18H6 emerges as the primary retrieval head across both task directions and more importantly, independent of the winning mechanism.

3.1 Direct logit attribution

To gain a better understanding of the role that the different attention heads play to get to the correct answer by writing its value to the residual stream at the final token position, we use DLA across all heads of layers 16-25. In short, DLA is used to decompose the model's output logit into a sum of per-component, e.g. per head, contributions. For our experiment, we calculate the difference vector between the correct answer logit direction and the average logit direction of all other candidates in the same prompt. Following standard DLA, we pull this vector back to the individual head dimension yielding the per head gain $\mathbf{g}^{(l,h)}$ (details in Appendix C). Intuitively $\mathbf{g}^{(l,h)}$ can be understood as the ideal write direction and $\Delta M_i^{(l,h)}$ as the contribution to the write of each individual source token. The contribution then decomposes into the attention score and the alignment of the value vector $\Delta M_i^{(l,h)} = \alpha_i^{(l,h)} \left(\mathbf{v}_i^{(l,h)\top} \mathbf{g}^{(l,h)}\right)$.

We show a summary of our results in the left column of Table 1. Here, L18H6 arises as the strongest retrieval head with 98% of its write attributed to the value of the single candidate token across both schema. Notably, we end up identifying another class of heads, which are not retrieving, but amplifying the retrieved signal further. We go more in detail on these in Appendix Section C and include a comprehensive breakdown of individual token positions in Tables 2 - 5.

3.2 Testing causality of the retrievers

DLA is a powerful screening tool, but only measures a component's direct contribution to the target direction and does not include indirect effects caused by later interaction with other components. For a causal test we apply a counterfactual patch at the input of layer 18. We calculate the average margin between the base answer and the strongest alternative under the counterfactual patch before and after patching. This comes out to +46.90 (box $\rightarrow$ object) and +36.88 (object $\rightarrow$ box). Following Wang et al. (2023), for each head in layers 18-25, we measure how much of that flip is undone by restoring a) the whole-head output, b) only rescaling the head's attention distribution within the 20 candidate tokens back to their base proportions while leaving all other positions and values patched, c) rescaling across all other positions. We then measure how much of the full restoration effect W is achieved just by rescaling the candidate attention weights C and compute the ratio C/W.

The center column of Table 1 shows that the previously identified heads through DLA also have a large causal effect. The strongest, L18H6 reverses over 36% ($W = 17.25$) of the average margin the counterfactual patch establishes and almost the entirety (95%) of that effect is explained by

Box → Object

Head	DLA Candidate	DLA Total	Causal Restoration (All) W	C	C/W%	W split by Mechanism Reflexive	Lexical	Positional
L18H6	3.74	3.80	17.25	16.35	**95%**	17.38	18.09	16.10
L22H4	3.10	3.28	7.40	5.72	**77%**	7.33	8.55	7.24
L19H1	2.74	3.21	8.73	5.66	**65%**	8.18	9.50	9.07

Object → Box

Head	DLA Candidate	DLA Total	Causal Restoration W	C	C/W%	W split by Mechanism Reflexive	Lexical	Positional
L18H6	3.69	3.70	21.54	20.80	**97%**	23.35	21.80	19.59
L23H2	2.79	3.04	5.35	5.61	**105%**	5.07	5.36	5.28
L19H1	2.62	2.66	12.88	11.48	**89%**	13.57	13.05	10.91

Table 1: Top three retriever heads filtered and ranked by total write with L18H6 emerging as strongest. **Left**: Direct logit attribution is heavily concentrated on the candidate token position (candidate and comma token). **Center**: Restoring only the attention distribution over candidate tokens (C) recovers 65%–105% of the whole-head restoration effect (W). **Right**: This restoration is consistent across eventual selected mechanism. **L18H6** write and causal effect is strongest and consistent across mechanisms.

shifting the attention of the candidate tokens back to the base distribution. This is independent of the winning mechanism (right column). In comparison, the amplifying heads show very low C/W, see Appendix Section D for a more complete breakdown and the analysis of all relevant heads. Taken together, we identify L18H6 as the strongest retriever, whose write is strongly aligned with the final target direction and strongly determines the final logit causally.

4 Isolating the Residual Stream Subspaces

Now that we have located the primary retrieval head, we investigate the residual stream to find what is driving this head's attention. The goal is to decompose the residual stream into these parts and later trace each component through L18H6's attention computation separately to observe their influence.

Towards this end, we follow the counterfactual-pair approach to identify linear representations (Park et al., 2024; Zou et al., 2023), constructing prompt pairs that differ in one variable: two facts trade positions (positional), the queried entity is swapped (lexical), or the answer entity is swapped (reflexive). Each class (e.g. position 0, apple-as-query, or magnet-as-answer) gets 16 pairs, for 8,192 prompts in total across both task formats. At the final token position, for each layer and mechanism, we compute the mean paired difference per class. Treating each mechanism as a categorical concept (Park et al., 2025), we define its subspace as the span of the class means.

This process yields three low-rank subspaces: effective rank[2] 3 for positional, and 10 and 13 for lexical and reflexive respectively. We examine the nature of these representations in more detail in Section 6. Here, we show that the subspaces together carry the effect of a full counterfactual patch and can be edited independently to redirect the model's answer.

4.1 Causal Sufficiency and Separability

We test the subspaces causally by first checking that they carry the effect of a full counterfactual patch, and then editing one at a time to selectively redirect the answer. Let $x \in \mathbb{R}^d$ be the residual stream at the final token position entering layer 18. We first split out a generic task component, g, by averaging over x for all prompts by task format. We write

[2]The participation ratio of the singular values $(\sum_i s_i^2)^2 / \sum_i s_i^4$ for singular values s_i of the class-mean matrix.

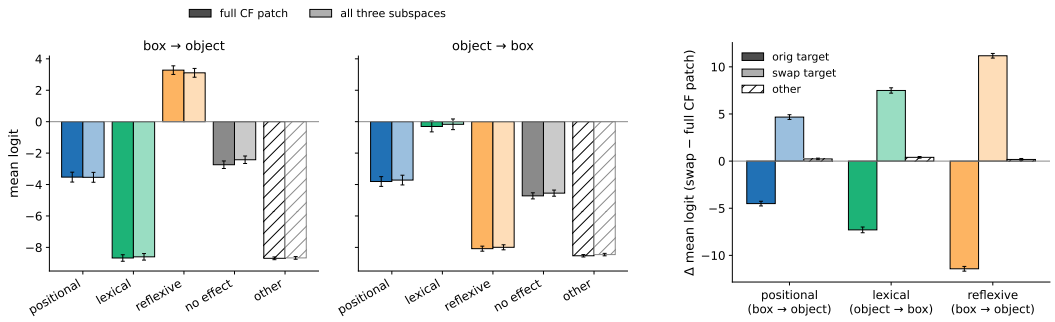


Figure 2: **Left:** Mean answer token logits under full counterfactual patch vs base prompt + all three subspaces patched in. **Right:** Mean answer logit delta between original counterfactual target and swap target per subspace.

$$x \;=\; g + x_{\text{pos}} + x_{\text{lex}} + x_{\text{ref}} + x_{\perp}, \tag{1}$$

where x_{pos} + x_{lex} + x_{ref} is the projection of $x - g$ onto the span of the three mechanism subspaces and $x_{\perp}$ is what remains.

We use 1,000 counterfactual prompt pairs (500 per task format) and decompose the layer 18 residual stream of both the base and counterfactual prompts as in Eq. 1. We then replace a base prompt's x_{pos}, x_{lex}, and x_{ref} with counterfactual prompt's, keeping g constant, and $x_{\perp}$ from the original. Figure 2 shows the mean final logits of each mechanism's target token, the base answer (no effect), and other tokens under each patch version. The subspace patch essentially reproduces the effect of the full counterfactual patch, selecting the same top answer token on 93.0% (box $\rightarrow$ object) and 94.4% (object $\rightarrow$ box) of prompts. The reverse, keeping the three mechanism components from the base prompt, with g and the counterfactual's $x_{\perp}$, returns to the base answer on 99.6% and 99.8% of prompts.

We then ask if the individual subspaces are independently manipulable. To examine this, we use a modification to the counterfactual patch method. We use the same 1,000 counterfactual prompts, then, one at a time, we subtract the class mean of the variable a given mechanism predicts and add in the class mean of a different target answer. In this way, we are able to selectively adjust a particular mechanism and leave the others unchanged. Figure 2 shows the resulting change in the final logit for the original target, the new swap target, and other candidates, for each mechanism in its dominant task format (Appendix B). In each case, we see the final logit shift from the original target to the swap target without substantially shifting the other candidates.

5 Additive Combination in QK Space

We now combine the results of Sections 3 and 4 and show how the mechanism subspaces in the residual stream drive the attention of the primary retrieval head to select an answer. Because the subspaces are additive in the residual stream, and the query key space is bilinear, we can decompose the attention logits directly. We follow Lee et al. (2026) in summing decomposed query contributions to attention logits, however, we want to understand the influence of elements in the residual stream on the attention, so we start from there instead of decomposing the query itself. We also split off generic task-related attention, g, and eliminate calculation order effects. Without splitting off g, general task attention is attributed to whichever mechanism query subspace overlaps it (in L18H6, most of it gets absorbed into the order-ID subspace under an uncentered contrastive covariance estimator, see Appendix E).

In this section, we show that each subspace contributes to its own target and that editing the subspace moves only that contribution. This suggests that multiple mechanism signals are present in the residual stream up until the retrieval head, at which point selection occurs via softmax over the votes.

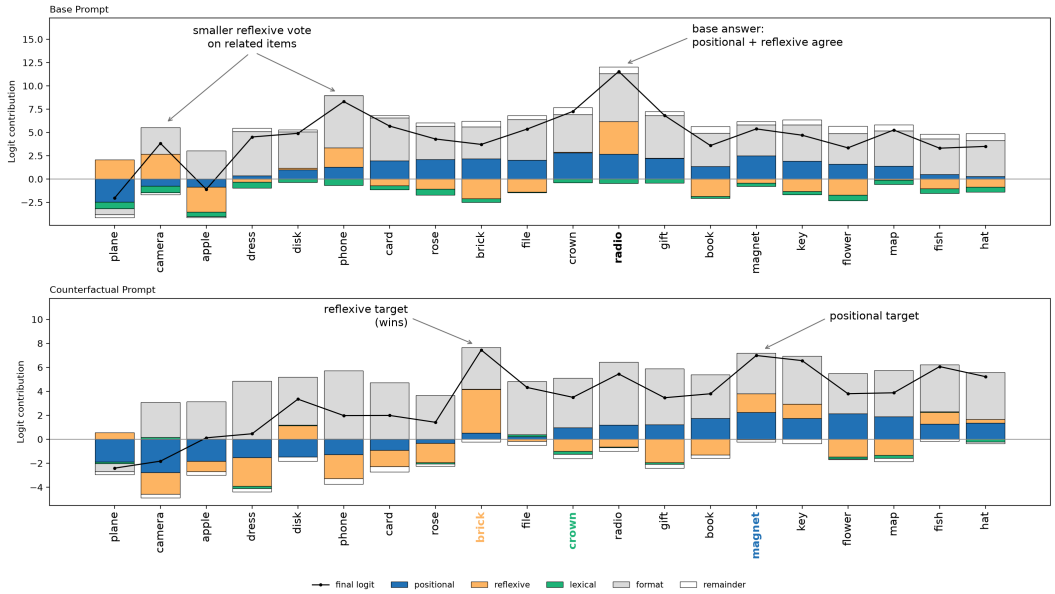
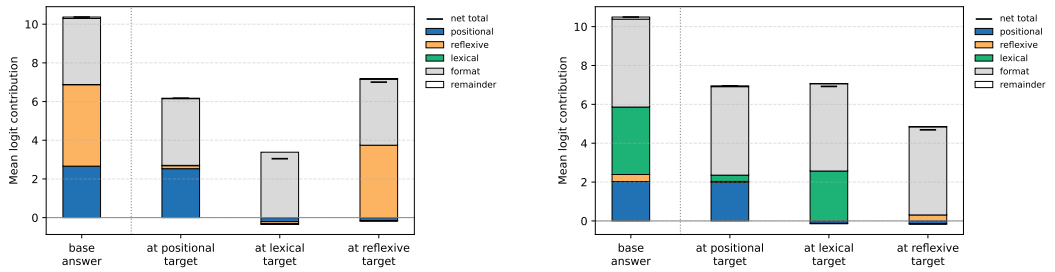


Figure 3: Attention decomposition of L18H6 for a single prompt. Answer tokens only.



(a) box → object

(b) object → box

Figure 4: Aggregated attention decomposition of L18H6 across target answer tokens in counterfactual prompts.

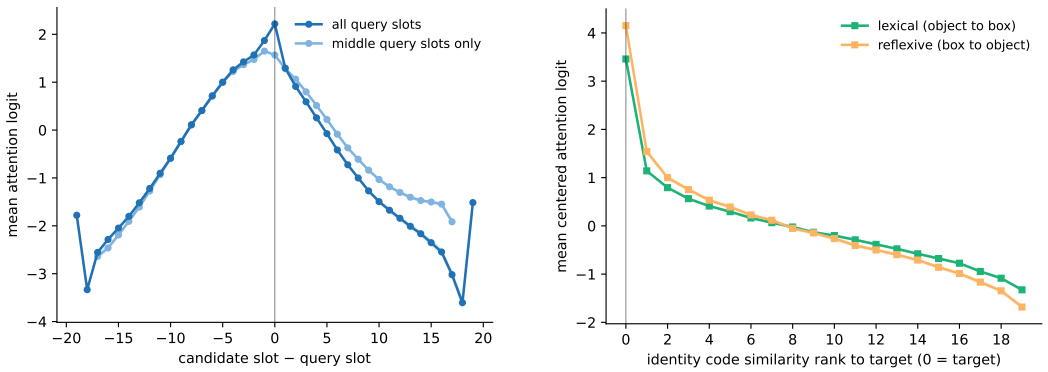
5.1 Attention Decomposition

We start with Eq. 1's decomposition of the final token residual stream entering layer 18, $x = g + x_{\text{pos}} + x_{\text{lex}} + x_{\text{ref}} + x_{\perp}$. To ensure order does not change the breakdown, we get x_{pos}, x_{lex}, and x_{ref} via a joint least squares fit of $x - g$ onto the span of the three mechanism subspaces at once. We can then pass this decomposition through to L18H6's attention computation by similarly decomposing the final token query vector, $q = W_Q x$. Let K be the stacked key vectors of the T prior tokens, so $K \in \mathbb{R}^{T \times d_{\text{head}}}$ and the attention computation is $\ell = Kq/\sqrt{d_{\text{head}}}$. Then for $c \in \{g, \text{pos}, \text{lex}, \text{ref}, \perp\}$, we have

$$q = q_g + q_{\text{pos}} + q_{\text{lex}} + q_{\text{ref}} + q_{\perp}, \qquad q_c = W_Q\, x_c, \tag{2}$$

$$\ell = \ell_g + \ell_{\text{pos}} + \ell_{\text{lex}} + \ell_{\text{ref}} + \ell_{\perp}, \qquad \ell_c = \frac{K q_c}{\sqrt{d_{\text{head}}}}. \tag{3}$$

By this method, we get the breakdown of the primary retrieval head's attention on each token by residual stream subspace contribution. The softmax of the reconstructed logits matches the model's actual attention computation to within 0.005 (max absolute difference; see Appendix F for handling of RMSNorm, RoPE, and soft cap). We use 1,000 examples, 500 per task format.



(a) Positional

(b) Content

Figure 5: **Attention logit contribution distribution.** (a) Diffuse positional gaussian-like distribution with exceptions at the poles, (b) content-matching logits nearly one-hot at target but graded by semantic similarity.

Across base and counterfactual prompts, the generic task component plus the three subspaces accounts for 99.7% of the attention logit variance. The generic task component drives attention towards the correct answer format for the particular task, with an average attention logit contribution of +4.5 for box letter tokens vs -2.0 for object tokens in the object → box task and +3.4 vs -1.3 for object vs box letter tokens in the box → object task. The mechanism subspaces also contribute according to expectations: in base prompts, both the positional and the relevant content-matching mechanism contribute to the target token additively. In counterfactual prompts, the contributions split, with the positional contribution strongest at the positional target and the lexical and reflexive subspace contributions strongest at their relative targets in their respective dominant task formats (Figures 4b and 4a). Consistent with counterfactual final logit results and expected behavior due to word order effects (Appendix B), the lexical subspace does not contribute significantly to the L18H6 attention logit in box → object tasks and vice versa for the reflexive subspace.

5.2 Mechanism Interventions

We rerun the targeted subspace patches described in Section 4.1 and observe the attention logit decomposition of L18H6. As expected, the subspace edits selectively move that mechanism's attention contribution to its intended target (Figure 12). Once the attention is moved, L18H6's DLA of the old mechanism target and new swapped target shift accordingly as well (Table 11).

5.3 Graded Readout

The positional mechanism does not vote for a single fact. Figure 3 shows the component-wise attention logits of L18H6 on the answer-format tokens of a single base prompt and its counterfactual. At the top we can see the diffuse contribution of the positional subspace centered at *radio*, while in the counterfactual, we can see it centered on *magnet*. Figure 5a shows this pattern in aggregate across prompts: Gaussian-like except for the effect of first and last items receiving sharper votes. This is the mechanistic counterpart of the Gaussian positional contribution to the final logit Gur-Arieh et al. (2025) inferred by fitting a causal model to the model's behavior. Here we observe this pattern, but upstream, in the attention logits of the primary retrieval head.

The content-matching mechanisms have previously been characterized as one-hot influences on the answer logits. While we do find their votes to be sharper than the positional vote, we find both lexical and reflexive subspace attention contributions to be graded as well. Figure 5b shows the pooled lexical and reflexive contributions to logits ranked by the similarity of each candidate's class mean to the target's. We see that there is a spike at the target, with a sharp fall off directly after, and from there the contribution falls monotonically with similarity rank. In Section 6 we show that this similarity structure is semantic.

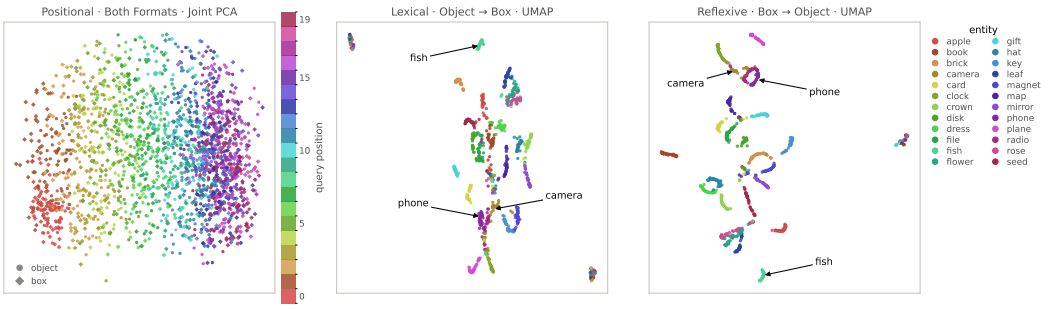


Figure 6: PCA and UMAP of final token residual streams at layer 18, selection of 24 categories of queried/answer objects only.

6 Mechanism Representations

In sections 4 and 5 we showed how the mechanism subspaces influence retrieval head behavior and drive final answer selection. Here we ask what they encode in order to do that. The positional subspace encodes fact order, in line with prior work. The lexical and reflexive subspaces are not as discrete as expected. They share a representational channel of the entity they use for their separate mechanisms (queried entity as key or answer entity as pointer), compressed along semantic lines.

6.1 Position

The positional mechanism subspace is effective rank 3 for the 20 item task and the first principal component direction is graded with position index. Figure 6 shows the principal component analysis (PCA) of the stacked residual streams of 2,000 ordinary prompts, position index is clearly aligned with PC1. This is consistent with prior findings like Dai et al. (2024) who performed a similar isolation. Interestingly, this is also consistent with Lee et al. (2026)'s findings which measured the positional subspace from within the query key space itself.

6.2 Content (Lexical and Reflexive)

While the lexical and reflexive mechanisms have been defined separately by the different references they use (queried entity for lexical, and answer entity for reflexive), we find that the subspaces they use to store their references overlap substantially. Both use the representation of the entity itself compressed along semantic lines.

Figure 6 shows UMAP of stacked final token residual streams at layer 18 (2,000 prompts, 24 objects). In the lexical dominant format (object $\rightarrow$ box), the vectors cluster by queried object. In the reflexive dominant format (box $\rightarrow$ object), they cluster by answer object. The top 10 directions of the subspaces have a mean principal cosine of 0.83, with random rank-10 spans at 0.06. We additionally find a 0.78 top 10 mean cosine similarity between the subspaces in a version of the tasks where the reflexive pointer is a box letter and the lexical key is an object, showing that the overlap is not simply due to both references being objects (Appendix H).

The shared channel carries compressed representations of the entities. The subspaces are low rank and grow sublinearly with vocabulary, with both subspaces effective rank 9 at 24 objects, and with rank 10 (lexical) and 13 (reflexive) at 82 objects, consistent with Lee et al. (2026)'s characterization in QK-space. The compression in the residual stream channel occurs along representational similarity lines. The cosine similarity structure of the 82 class means correlates with the same objects' residual stream representations in ordinary sentences to 0.53 for lexical and 0.60 for reflexive, compared with 0.26 and 0.31 for the token embeddings (Appendix I). Errors show the same structure: 44% of mistakes fall on the semantic nearest neighbor in the lexical dominant task format (object $\rightarrow$ box) and 27% in the reflexive dominant format (box $\rightarrow$ object) with chance at 5%. Thus, rather than using an arbitrary identity tag when using these two mechanisms, the model content-matches on a compressed version of the concept itself: the queried entity for the lexical mechanism, and the answer entity for the reflexive mechanism.

This overlap does not contradict Section 4.1 because those experiments test each content mechanism only in the format where it is dominant and the other contributes very little. How the two mechanisms might share the channel when both are used, and whether a swap on one would shift another, we leave to future work.

7 Related Work

Retrieval heads There are many established patterns of attention heads copying information from earlier positions, including induction heads, name mover heads, attribute extraction heads and long context retrieval heads (Elhage et al., 2021; Olsson et al., 2022; Wang et al., 2023; Geva et al., 2023; Wu et al., 2025a). We identify our retrieval heads with direct logit attribution (Elhage et al., 2021) and confirm them with activation patching, separating attention pattern from its values as in Wang et al. (2023). Our primary retrieval head, Gemma 2 2B's L18H6, has also been found specifically to be involved in retrieval on an abstract rule induction task (Yang et al., 2025), suggesting its role may be more general than this task family.

Variable binding Feng & Steinhardt (2024) showed that models tag entities and attributes with additive binding vectors, Dai et al. (2024) characterized these as low-rank and graded with fact order. The Prakash et al. (2024; 2025) line of work further characterized these as lookbacks and traced the circuits involved in retrieval. Gur-Arieh et al. (2025) showed that in longer contexts with more than a few items in a list, other mechanisms emerge, some of which we examined. Wu et al. (2025b) found that in a synthetic setting, a positional heuristic arises in training before a systematic content-matching approach to solving a symbolic program.

Subspaces and QK space In this work we estimate the mechanism subspaces using counterfactual prompt pair differences (Zou et al., 2023; Panickssery et al., 2024). Geiger et al. (2024) established another way of identifying causal subspaces with Distributed Alignment Search, and Prakash et al. (2024) use desiderata-based component masking. We use difference-based estimation because it doesn't require optimization and defines the subspace directly rather than matching to a causal model. Lee et al. (2026) isolate subspaces specifically in the query-key space of attention heads with their contrastive covariance method. They isolate QK subspaces involved in the positional and lexical mechanisms and split attention logits into their component parts. Their method isolates subspaces in the query while we trace already identified residual stream subspaces. We use the same method of decomposing the resultant attention logit computation, however, we also split out general task-related attention in the residual stream whose contribution otherwise gets absorbed into the mechanism subspaces.

8 Discussion

In this work, we located the subspaces of the residual stream used by three retrieval mechanisms, showed they are individually steerable, and that they combine additively in the query of a single primary retrieval head. Here we discuss limitations, potential future directions and broader implications.

We focused on the final token position, which is causal and sufficient to impact output but doesn't answer the question of how the signal was created and moved to the final token. The boxes task is a simple, constrained task, with only two entity types (boxes and objects). How the attention decomposition works in higher entity or more naturalistic settings we leave to future work. Additionally, we examined only two models of two model families. While the general findings replicate from Gemma 2-2B to Llama 3.1-8B (Appendix A), we already see less concentration in terms of retrieval head function. It is not clear if this is model family related or related to scale. More work needs to be done to characterize the nature of the lexical and reflexive shared content-matching channel. In future work, we may examine the overlap and differences of those, with potential more broad implications for bag-of-words vs role-filler binding structure.

Broadly, this work is one example of how a model implements redundant mechanisms and reconciles competing hypotheses: by parallel signals in the residual stream and with softmax as the arbiter. Whether this pattern holds elsewhere is open.

9 AI USE STATEMENT

AI was used under the authors' direction for help with research ideation, feedback on methodology and experiments, code writing and implementation, as well as a supplement to standard literature review and for feedback on paper structure and clarity of narrative. All text was written by the authors.

10 REPRODUCIBILITY STATEMENT

Code available at `https://anonymous.4open.science/r/binding-mechanisms`.

REFERENCES

Qin Dai, Benjamin Heinzerling, and Kentaro Inui. Representational analysis of binding in language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

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. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

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

Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1–2):3–71, 1988.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pp. 160–187. PMLR, 2024. arXiv:2303.02536.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, et al. Gemma 2: Improving open language models at a practical size, 2024. URL `https://arxiv.org/abs/2408.00118`.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. arXiv:2304.14767.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

Yoav Gur-Arieh, Mor Geva, and Atticus Geiger. Mixing mechanisms: How language models retrieve bound entities in-context. *arXiv preprint arXiv:2510.06182*, 2025.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3835–3855, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.213. URL `https://aclanthology.org/2023.acl-long.213/`.

Andrew Lee, Yonatan Belinkov, Fernanda Viégas, and Martin Wattenberg. Decomposing query-key feature interactions using contrastive covariances. *arXiv preprint arXiv:2602.04752*, 2026.

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. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. arXiv:2312.06681.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. arXiv:2311.03658.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2406.01506.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *International Conference on Learning Representations*, 2024.

Nikhil Prakash, Natalie Shapira, Arnab Sen Sharma, Christoph Riedl, Yonatan Belinkov, Tamar Rott Shaham, David Bau, and Atticus Geiger. Language models use lookbacks to track beliefs. *arXiv preprint arXiv:2505.14685*, 2025.

Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1–2):159–216, 1990. doi: 10.1016/0004-3702(90)90007-M.

Anne Treisman. The binding problem. *Current Opinion in Neurobiology*, 6(2):171–178, 1996. doi: 10.1016/S0959-4388(96)80070-5.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *International Conference on Learning Representations*, 2025a. arXiv:2404.15574.

Yiwei Wu, Atticus Geiger, and Raphaël Millière. How do transformers learn variable binding in symbolic programs? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025b. URL https://proceedings.mlr.press/v267/wu25j.html.

Yukang Yang, Declan Iain Campbell, Kaixuan Huang, Mengdi Wang, Jonathan D. Cohen, and Taylor Whittington Webb. Emergent symbolic mechanisms support abstract reasoning in large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70515–70549. PMLR, 2025. URL https://proceedings.mlr.press/v267/yang25c.html.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt

Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

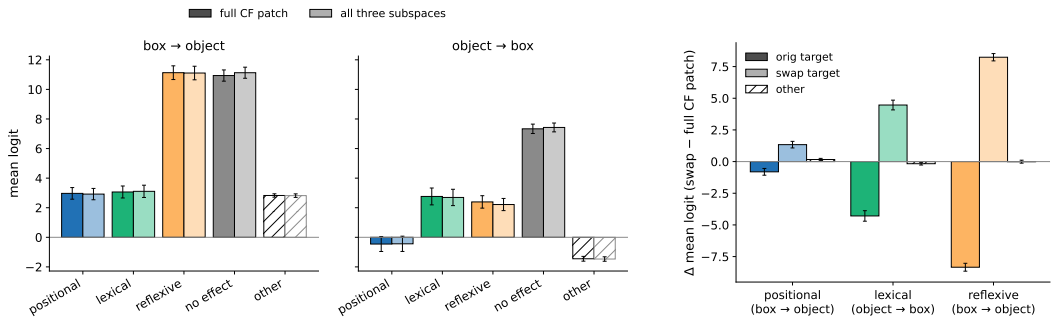
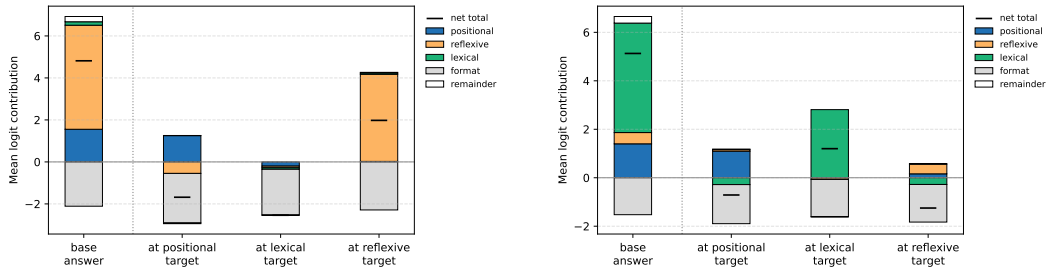


Figure 7: **Left**: Mean answer token logits under full counterfactual patch vs base prompt + all three subspaces patched in for Llama 3.1-8B. **Right**: Mean answer logit delta between original counterfactual target and swap target per subspace in Llama 3.1-8B.



(a) box → object

(b) object → box

Figure 8: Aggregated attention decomposition of L17H24 in Llama 3.1-8B across target answer tokens in counterfactual prompts.

A Replication in Llama 3.1-8B

We conduct all experiments from the prior sections on Llama 3.1 8B using a 24 object vocabulary. The circuit identification method yields a collection of attention heads involved in the final retrieval from layer 16 to 22. While the computation appears to be less concentrated in one head than in Gemma 2-2B, L17H24 places 60% of its non-BOS attention on the answer token, against L18H6 Gemma 2-2B's 74% and attends to all three counterfactual targets. Later heads appear to be more reflexive-specific and follow the reflexive target almost exclusively. We focus on layer 17 and L17H24 for our experiments.

We estimate the mechanism subspaces using the paired difference method and find them consistent with the subspaces found in Gemma 2-2B and find them to be sufficient under the same methodology as Section 4 with accuracy going from 99.4% $\rightarrow$ 99.4% for the box$\rightarrow$object format, and 99.8% $\rightarrow$ 99.8% object$\rightarrow$box after replacing layer 17 with g plus the joint subspaces. The positional subspace is effective rank 3 with the first principal component graded with fact index, while lexical and reflexive are effective rank 11 and 10 respectively. The content-matching subspaces also overlap in Llama 3.1-8B as in Gemma, with the top 10 spans having mean principal cosine 0.73 where random rank-10 spans give 0.04. Targeted subspace swaps on counterfactual patches roughly swap the final logit values of the original mechanism target vs the new swap target, as expected, and leave the other targets in place (Figure 7).

For the attention decomposition, we use Equations 1–3, omitting only the Gemma 2 soft cap, and achieve reconstruction accuracy to within 0.003. The generic task component again differentiates between appropriate answer token format with a margin of 2.2 - 4.3 logits depending on the task. Mechanism subspaces contribute as expected, with the positional contribution a diffuse signal roughly centered on the target, with content-matching subspaces with sharper spikes at their target, sometimes with runner-up spikes at semantically similar tokens.

Llama's accuracy is higher than Gemma's and is largely saturated, making an error analysis intractable, however we see similar graded attention readouts within the retrieval head and runner-up answers by final logit have a high semantic nearest-neighbor rank.

The primary difference across the models is concentration of retrieval in one primary retrieval head. We do still find a dominant head that reads all three mechanisms, L17H24, however there also appear to be reflexive mechanism specialists in later layers, contributing to a band of heads involved in final retrieval.

B Word Order Influences the Mechanism Mix

Utilizing the full-residual stream counterfactual patch at layer 18, we conduct a two-by-two experiment in order to investigate how the different mechanisms depend on the task and query format. For that, we vary the query format: box → object (What does Box X contain?) and object → box (Which box is the item in?). And, the task format, either naming the object first (the item is in Box X) or the box (in Box X is the item). We keep facts, order and question identical across the base and the counterfactual and only vary the query and fact format as described. We find that a strong predictor for whether dominant mechanism ends up being reflexive or lexical is the combination of what the query asks for, and whether that entity appears on the left or on the right of its pair. Figure 9 demonstrates this across a dataset of 3,000 examples. Pooled across formats, the reflexive mechanism is more than 3 times more likely to be dominant when the answer is left of the cue in the task (53.5% to 16.0%) compared to when it's on the right. The reverse is true for the lexical mechanism (1.8% to 32.2%). Figures 10, 11 show this same phenomena in more detail, split by position and varied counterfactual patch layer.

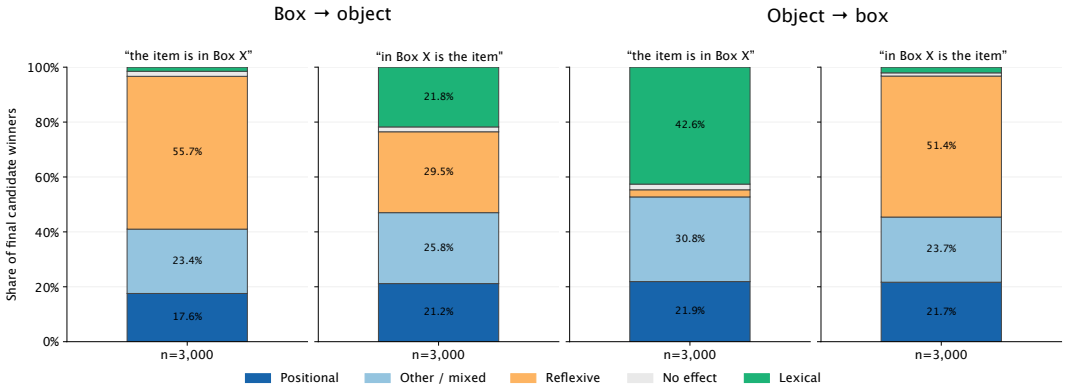


Figure 9: Distribution of dominant mechanism across varied task and query formats after counterfactual patching at the residual stream of Layer 18. Left: Query format "What does Box X contain?". Right: Query format "Which box is the item in?".

C Direct Logit Attribution

$$\mathbf{g}^{(l,h)} = \underbrace{\left(W_O^{(l,h)}\right)^\top}_{\text{Head output projection}} \cdot \underbrace{s_{\text{post}}^{(l)}\left(1+\mathbf{w}_{\text{post}}^{(l)}\right)}_{\text{Layer RMSNorm}} \odot \underbrace{s_{\text{final}}\left(1+\mathbf{w}_{\text{final}}\right)}_{\text{Final RMSNorm}} \odot \underbrace{\left(W_U[c^*] - \frac{1}{|\mathcal{C}|-1}\sum_{c\neq c^*} W_U[c]\right)}_{\text{Unembedding margin difference } \mathbf{r}} \tag{4}$$

Equation 4 formally explains how the per head gain $\mathbf{g}^{(l,h)}$ is computed. For that, we start by projecting the difference of the correct answer logit direction c^* and the average logit direction of all other candidate readouts in the same prompt and obtain the margin difference $\mathbf{r}$. We make use of the frozen scalars s_{final} and s_{post}^{l} to propagate the difference $\mathbf{r}$ back to the individual layer. There, we

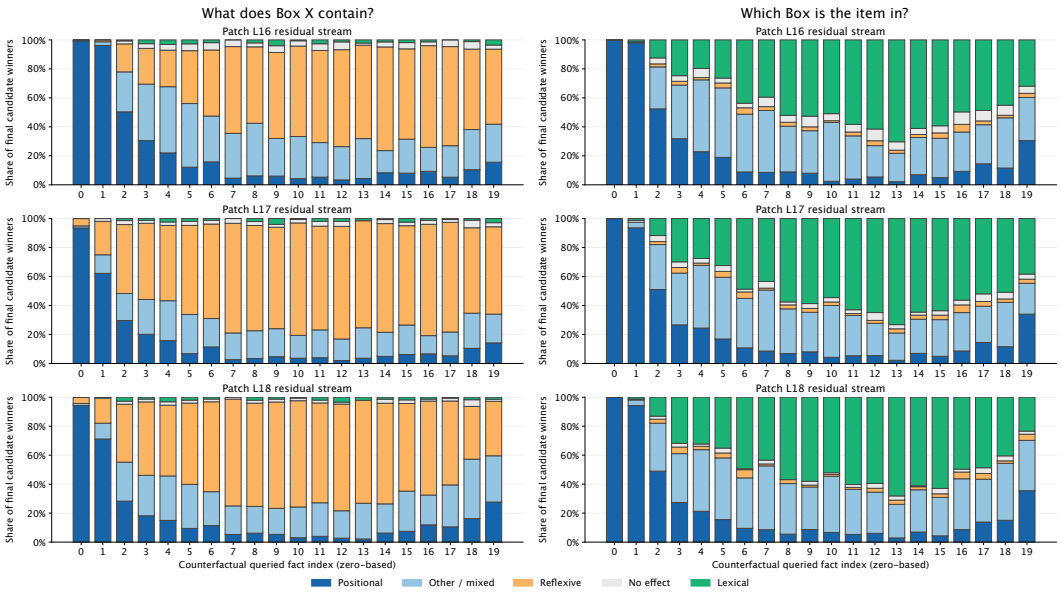


Figure 10: Distribution of dominant mechanism split by positions and with prompt order "the item is in Box X". Reflexive mechanism much more dominant when answer on left of queried entity. Lexical mechanism is the reverse.

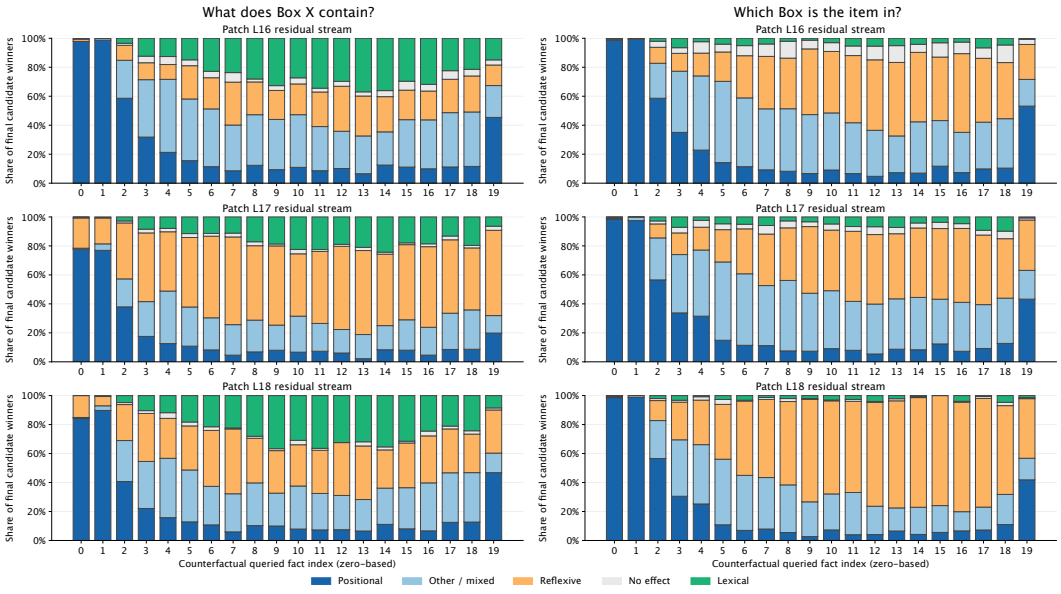


Figure 11: Distribution of dominant mechanism split by positions and with prompt order "in Box X is the item". Reflexive mechanism more dominant when answer on left of queried entity. Lexical mechanism is the reverse.

project through the transpose of the respective head's output matrix $(W_O^{(l,h)})^T$. The resulting term $\mathbf{g}^{(l,h)}$ can be intuitively understood as the ideal write direction maximizing $\mathbf{r}$.

$$\Delta M^{(l,h)} = \sum_i \underbrace{\alpha_i^{(l,h)}}_{\text{Attention}} \cdot \underbrace{\left(\mathbf{v}_i^{(l,h)\top} \mathbf{g}^{(l,h)} \right)}_{\text{Value Alignment}} \tag{5}$$

For a single head, the output is an attention-weighted sum of the individual value vectors $\sum_i \alpha_i \mathbf{v}_i$. So as showcased in Equation 5, a single head's total effect on the margin $\Delta M^{l,h}$ can be split into the individual source token contributions. This term depends on which position the head attends to α_i and how strongly this token's values align with the correct answer direction $\mathbf{v}_i^T \mathbf{g}$.

For space and legibility reasons, we report all active heads with a total write greater than 0.3. Tables 2, 3 show this for the task of retrieving the given item in a Box. The majority of the write for retriever heads is coming from the candidate token, which represents the concrete item being retrieved. For some, additional signal comes from the following single comma token as well (Table 2). L18H6 together with L22H4 and L19H1 shapes the majority of that retrieval. Retrieval appears to be mainly taking place throughout Layers 18-22.

Head	Candidate	Comma	Question	-8	-7	-6	-5	-4	-3	-2	-1	Rest	Total
L18H6	3.71	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	3.80
L22H4	2.28	0.82	0.00	0.06	0.00	0.00	0.00	0.05	0.00	0.00	0.01	0.06	3.28
L19H1	1.68	1.06	0.00	0.02	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.42	3.21
L22H5	1.09	0.35	0.00	0.08	0.00	0.00	0.00	0.05	0.00	0.00	0.01	0.09	1.67
L21H5	-0.64	-0.22	0.00	-0.04	0.00	-0.02	0.00	-0.05	0.00	0.00	-0.04	-0.02	-1.03
L24H7	0.62	0.17	0.01	0.10	0.01	0.01	0.00	0.06	0.00	0.00	0.02	0.02	1.02
L20H7	0.84	0.13	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.03	1.02
L22H2	0.70	0.16	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.04	0.92
L16H2	-0.37	-0.09	0.00	-0.02	0.00	-0.01	0.00	-0.04	0.00	0.00	-0.05	-0.23	-0.81
L16H4	0.33	0.10	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.21	0.75
L23H5	-0.38	-0.20	0.00	-0.02	0.00	0.00	0.00	-0.06	0.00	0.00	-0.01	-0.03	-0.70
L18H7	-0.64	-0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	-0.68
L18H5	0.54	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.59
L18H3	0.28	0.03	0.02	0.01	0.00	0.00	0.00	0.16	0.00	0.00	0.01	0.08	0.59
L19H0	0.36	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.53
L22H3	0.32	0.07	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.42
L17H3	0.06	0.15	0.00	-0.00	0.00	0.00	0.00	-0.00	0.00	0.00	0.00	0.14	0.35
L23H7	0.24	0.07	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.34
L18H2	0.27	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.03	0.33
L16H5	0.23	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.30

Table 2: DLA computation split by candidate head for all heads classified as mainly retrieving the value from the candidate token(s). Ranked by absolute total direct logit attribution. Box → object ("What does Box X contain?"). Most retrievers get the majority of their write directly from the candidate token, for some it's split with the neighboring comma token. Not all write in the positive margin direction.

For the heads amplifying the signal, this happens primarly at token indices -8, -6, -4 and -1. These correspond to the "?", ":", and the two \n tokens because of the way the end of our prompts are formatted by Gemma 2: "What does Box D contain? Answer: <end_of_turn>\n<start_of_turn>model\n." While we only explore the retrieval more directly at the final token, we hypothesize that this is largely mirrored at the other identified positions.

Head	Candidate	Comma	Question	-8	-7	-6	-5	-4	-3	-2	-1	Rest	Total
L20H3	0.41	0.27	0.02	0.63	0.00	0.12	0.00	1.98	0.00	0.00	0.26	0.28	3.97
L20H2	-0.30	-0.94	-0.01	-0.55	0.00	-0.27	0.00	-0.52	0.00	0.00	-0.80	-0.20	-3.59
L24H3	0.00	0.03	0.01	0.28	0.03	0.83	0.01	0.51	0.00	0.16	0.54	0.00	2.40
L25H4	0.02	0.02	0.01	0.33	0.02	0.29	0.00	0.49	0.00	0.12	0.76	0.02	2.08
L24H2	0.00	-0.02	0.00	-0.24	-0.03	-0.21	-0.01	-0.12	0.00	-0.01	-0.83	-0.03	-1.50
L21H1	0.00	0.01	0.01	0.09	0.00	0.17	0.00	0.24	0.00	0.02	0.60	0.01	1.15
L25H5	-0.01	-0.01	-0.01	-0.08	-0.02	-0.08	-0.01	-0.04	0.00	-0.03	-0.78	-0.02	-1.09
L23H3	0.08	0.02	0.00	0.12	0.01	0.20	0.00	0.36	0.00	0.02	0.18	0.02	1.01
L25H2	0.00	0.02	0.01	0.15	0.01	0.14	0.00	0.26	0.00	0.02	0.11	0.01	0.73
L25H6	0.00	0.00	0.00	0.06	0.00	0.04	0.00	0.08	0.00	0.07	0.31	0.00	0.56
L25H7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.45	0.01	0.46
L21H0	0.07	0.02	0.01	0.04	0.00	0.03	0.00	0.08	0.00	0.00	0.06	0.02	0.33

Table 3: DLA computation split by candidate head for all heads classified as mainly amplifying the signal that's already present. Ranked by absolute total direct logit attribution. Box → object ("What does Box X contain?"). Amplifiers have the majority of their write direction coming from token indices -1, -4, -6 and -8. Not all write in the positive margin direction.

Tables 4, 5 provide the DLA results for the task of retrieving the correct Box given the item. The previous results mirror. The only thing that changes is that the prompt format includes an additional "Box" token before the "Answer" token, and we also see additional attention of the amplifiers on this position (-9).

Head	Candidate	Comma	Question	−9	−8	−7	−6	−5	−4	−3	−2	−1	Rest	Total
L18H6	3.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	3.70
L23H2	2.79	0.00	0.00	0.08	0.00	0.00	0.02	0.00	0.10	0.00	0.00	0.04	0.01	3.04
L19H1	2.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.01	0.00	2.66
L20H2	-0.59	-0.01	0.00	-0.18	-0.10	-0.01	-0.19	0.00	-0.21	0.00	-0.01	-0.35	0.00	-1.65
L22H3	1.17	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.00	1.21
L18H7	-0.52	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.51
L18H4	0.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.47
L24H7	0.31	0.00	0.00	0.03	0.05	0.03	0.00	0.00	0.03	0.00	0.01	0.01	-0.01	0.46
L22H4	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.45

Table 4: DLA computation split by candidate head for all heads classified as mainly retrieving the value from the candidate token(s). Ranked by absolute total direct logit attribution. Object → box ("Which box is the item in?").

Head	Candidate	Comma	Question	−9	−8	−7	−6	−5	−4	−3	−2	−1	Rest	Total
L25H4	0.01	0.00	0.01	0.38	0.58	0.08	0.29	0.01	0.52	0.00	0.26	1.00	0.00	3.14
L24H3	0.00	0.00	0.00	0.17	0.64	0.09	0.24	0.00	0.34	0.00	0.18	0.53	0.01	2.20
L24H2	-0.01	0.00	0.00	-0.19	-0.52	-0.06	-0.32	-0.01	-0.13	0.00	-0.02	-0.52	-0.01	-1.79
L25H5	-0.01	0.00	-0.01	-0.12	-0.36	-0.06	-0.17	-0.01	-0.06	0.00	-0.06	-0.80	0.00	-1.66
L21H1	0.00	0.00	0.00	0.04	0.43	0.01	0.11	0.00	0.11	0.00	0.01	0.36	0.00	1.07
L20H3	0.19	0.00	0.00	0.12	0.06	0.00	0.03	0.00	0.15	0.00	0.00	0.07	-0.01	0.61

Table 5: DLA computation split by candidate head for all heads classified as mainly amplifying the signal that's already present. Ranked by absolute total direct logit attribution. Object → box ("Which box is the item in?").

D Causal Analysis

Tables 6, 7 provide the full causal results for the earlier identified retrieval heads for the Box → Object task. Across all heads, the causal strength stays roughly equal, independent of which mechanism is selected as target.

Tables 8, 9 mirror the observations for the other task format.

E Comparison with Contrastive QK Covariance

We apply the order-ID estimator of Lee et al. (2026) to L18H6 of Gemma 2-2B on the object → box format with 20 facts. Following their recipe, each of 1,000 prompts uses the same set of 20 objects in a shuffled order, the same queried object, and random box letters. The query q is the final token's post-RoPE query and the keys are the post-RoPE keys at the box-letter tokens. The contrastive covariance is $\Delta C = \mathbb{E}[q\,k^{+\top}] - \mathbb{E}[q\,k^{-\top}]$, where k^+ is the key of the correct box and k^- ranges over the other 19 boxes. As in their method, queries and keys are not centered. We take the top three left singular vectors U of ΔC; their criterion of 99% of the squared Frobenius norm also gives rank 3, in line with the rank 2–3 they report. On 300 held-out standard prompts we compute order-ID logits from $q_{\text{order}} = UU^\top q$, as in their decomposition with order ID projected first, and split q_{order} into the part from q_g, the query image of g computed as in Eq. 2, and the part from $q - q_g$.

The top left singular vector of ΔC has cosine 1.00 with the mean query $\bar{q}$. Writing $q = \bar{q} + (q - \bar{q})$, the rank-one term $\bar{q}\,\mathbb{E}[k^+ - k^-]^\top$ accounts for 92% of the Frobenius norm of ΔC. The three-dimensional order-ID subspace contains 83% of the squared norm of q_g (1.2% for a random subspace of the same dimension) and 54% of the squared norm of our positional query component. Table 10 splits the order-ID logits. The part from q_g is the same on the target box and on the other boxes and negative on object tokens, the same token-type pattern as our format component. The difference between the target box and the other boxes comes from $q - q_g$ (1.8; q_g contributes -0.1) and is close to our positional component (2.2).

	Causal Restoration (All)			Causal W by Mechanism		
Head	W	C	C/W %	Reflexive	Lexical	Positional
L18H6	17.25	16.35	95%	17.38	18.09	16.10
L19H1	8.73	5.66	65%	8.18	9.50	9.07
L22H4	7.40	5.72	77%	7.33	8.55	7.24
L20H7	3.72	3.08	83%	3.69	3.87	3.48
L22H5	3.48	2.52	72%	3.39	3.00	3.89
L21H5	-2.32	-1.27	55%	-2.61	-1.57	-2.03
L18H5	2.10	1.19	57%	1.92	3.01	1.91
L18H3	2.03	0.81	40%	2.15	2.44	1.62
L18H7	-1.86	-1.72	93%	-1.80	-1.37	-1.65
L23H5	-1.84	-0.88	48%	-1.86	-2.07	-1.84
L22H2	1.74	1.09	63%	1.61	1.86	1.92
L18H2	1.65	0.93	57%	1.59	2.46	1.49
L20H6	-1.03	-0.62	60%	-0.99	-1.04	-1.03
L19H0	1.01	0.36	36%	0.93	1.23	1.17
L22H3	0.87	0.63	72%	0.85	1.07	1.03
L24H7	0.79	0.52	66%	0.83	0.94	0.69
L23H7	0.58	0.37	63%	0.55	0.47	0.67

Table 6: Causal restoration for Box → Object. Ranked by absolute whole-head restoration effect $|W|$, with W broken down across the three winning mechanisms.

	Causal Restoration (All)			Causal W by Mechanism		
Head	W	C	C/W %	Reflexive	Lexical	Positional
L20H3	9.80	1.31	13%	9.72	11.00	9.38
L20H2	-7.47	-0.77	10%	-7.32	-7.76	-7.12
L24H3	4.91	-0.01	-0%	4.94	4.62	5.00
L21H1	4.05	0.02	1%	4.15	4.03	3.78
L25H4	3.26	0.03	1%	3.24	3.63	3.35
L24H2	-2.90	0.00	-0%	-2.96	-2.73	-2.85
L23H3	1.68	0.12	7%	1.65	1.39	1.81
L25H5	-1.61	-0.01	1%	-1.63	-1.72	-1.63
L25H6	1.14	-0.00	-0%	1.18	1.08	1.09
L25H2	1.05	0.03	3%	1.02	1.16	1.14
L21H0	0.74	0.24	33%	0.71	0.69	0.78
L25H7	0.43	-0.00	-0%	0.45	0.63	0.31

Table 7: Causal restoration for Box → Object, amplifier heads. Ranked by absolute whole-head restoration effect $|W|$, with W broken down across the three winning mechanisms.

	Causal Restoration (All)			Causal W by Mechanism		
Head	W	C	C/W %	Reflexive	Lexical	Positional
L18H6	21.54	20.80	97%	23.35	21.80	19.59
L19H1	12.88	11.48	89%	13.57	13.05	10.91
L23H2	5.35	5.61	105%	5.07	5.36	5.28
L20H2	-3.20	-1.77	55%	-3.48	-3.14	-2.88
L18H4	1.98	1.71	86%	1.91	1.97	1.79
L22H3	1.68	1.73	103%	1.62	1.71	1.74
L22H4	1.60	1.73	108%	1.70	1.61	1.42
L18H7	-0.91	-1.00	110%	-1.02	-0.79	-0.71
L20H6	-0.70	-0.70	100%	-0.73	-0.70	-0.57
L24H7	0.45	0.36	81%	0.46	0.48	0.41
L18H5	0.31	0.15	48%	0.68	0.21	0.19

Table 8: Causal restoration for Object → Box. Ranked by absolute whole-head restoration effect $|W|$, with W broken down across the three winning mechanisms.

	Causal Restoration (All)			Causal W by Mechanism		
Head	W	C	C/W %	Reflexive	Lexical	Positional
L24H3	6.20	0.00	0%	6.26	6.37	5.93
L25H4	5.15	0.02	0%	5.07	5.23	5.10
L21H1	3.53	0.00	0%	3.52	3.55	3.36
L24H2	-3.07	-0.02	1%	-3.12	-3.16	-2.98
L25H5	-2.57	-0.01	0%	-2.49	-2.63	-2.57
L20H3	2.18	0.71	32%	2.20	2.08	1.97

Table 9: Causal restoration for Object → Box, amplifier heads. Ranked by absolute whole-head restoration effect $|W|$, with W broken down across the three winning mechanisms.

$\mathbb{E}[k^+ - k^-]$ is not zero because the queried object is fixed: the correct box is always the box bound to the same object, and we attribute the difference to box-letter keys carrying the identity of their object. Fact positions are balanced, so this is not an order imbalance. The norm of $\mathbb{E}[k^+ - k^-]$ is the same at 50 and 1,000 prompts (5.6 and 5.4). When the queried object varies across prompts, $\mathbb{E}[k^+ - k^-]$ shrinks with the number of prompts (1.3 at 50, 0.3 at 1,000), because keys do not depend on the question, and the share of q_g in the subspace falls from 0.90 to 0.24. With the queried object fixed, centering the queries before computing ΔC lowers the share to 0.08. We ran the estimator on our head and prompts, not on the heads and prompts of Lee et al. (2026) in Llama 3.1-8B. The result shows that the uncentered estimator absorbs the shared query; it does not show what their published subspaces contain.

F Attention Decomposition Detail

We handle the break in linearity from RMSNorm by computing the scale $s(x) = 1/\sqrt{\text{mean}(x^2) + \epsilon}$ once for the full vector x and applying that scale to each component so $\text{norm}(x_c) = x_c \odot s(x) \odot (1 + w)$ where w is the norm's learned weight and the components sum to $\text{norm}(x)$. W_Q and RoPE are linear, so they can also be applied to each component separately, where $q_c = \text{RoPE}(W_Q \,\text{norm}(x_c))$. Gemma 2 applies a soft cap $50 \tanh(\ell/50)$ to the logits before the softmax. We report pre-cap logits, sum the components and cap the total before the softmax.

Component	Target box	Other boxes	Objects	Target − other
Order ID (Lee et al., 2026)	4.62	2.95	−2.45	1.66
from q_g	2.98	3.08	−2.42	−0.11
from $q - q_g$	1.64	−0.13	−0.03	1.77
Ours: format (g)	4.49	4.54	−1.95	−0.05
Ours: positional	2.11	−0.09	−0.01	2.20

Table 10: Mean pre-softmax attention logit contributions in L18H6 (Gemma 2-2B), object → box, 300 prompts. Order ID uses the uncentered contrastive covariance estimator of Lee et al. (2026) at rank 3, fit on 1,000 prompts with a fixed queried object.

G L18H6 Under Mechanism Interventions

Figure 12 shows L18H6's attention logit decomposition at the original and the swap target before and after subspace swap over 500 prompts per task format. The attention moves as expected as the computation is linear. Figure 13 and Table 11 show the head's output DLA.

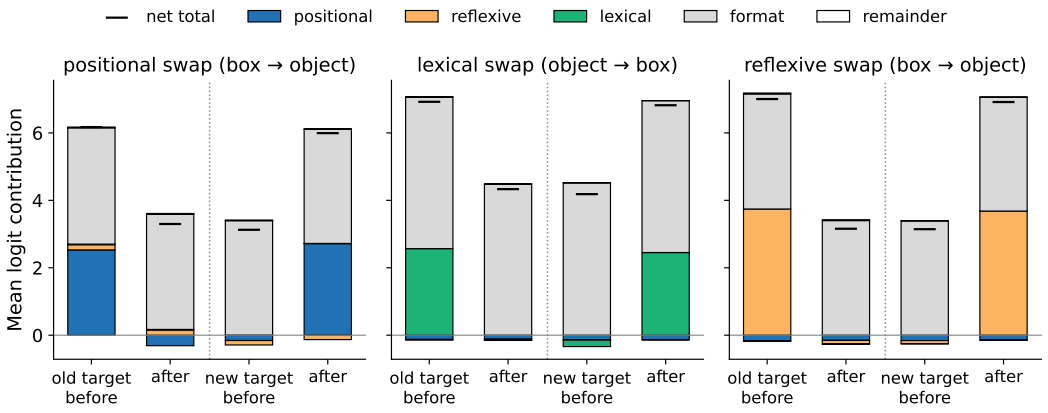


Figure 12: L18H6 component logits at the original and the swap target before and after subspace swap.

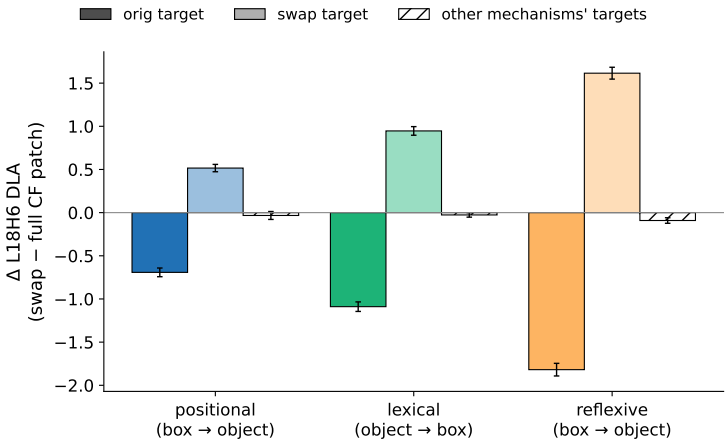


Figure 13: Change in L18H6's direct logit attribution (swap − full counterfactual patch) on the original target, the swap target and the other two mechanisms' targets.

		Old target		New target	
Format	**Swap**	Before	After	Before	After
box → object	positional	0.61	−0.08	−0.15	0.36
	lexical	−0.12	−0.11	−0.17	−0.17
	reflexive	**1.74**	−0.08	−0.12	**1.50**
object → box	positional	0.67	0.04	−0.08	0.27
	lexical	**1.11**	0.02	−0.11	**0.84**
	reflexive	−0.08	−0.13	−0.10	−0.05

Table 11: Direct logit attribution of L18H6 (average across 500 prompts) onto the old mechanism target and the swap target, before (full counterfactual patch) and after the targeted subspace swap at layer 18.

H Content Subspace Overlap Across Key Variations

The overlap between reflexive and lexical subspaces is not restricted to prompt conditions where the lexical key is an object and the reflexive pointer is an object. We use the reversed-order prompts from Section B and compare the measured lexical and reflexive subspaces when the key and pointer references are box letters. Table 12 shows the rank-10 mean principal cosine between lexical and reflexive spans at layer 18 for the four combinations of key types. All four are at least 0.78 with random at 0.06. Across word order variants, there may be some confounding from upstream context differing, but overall these results support the claim that the shared channel is not specific to the object vocabulary.

	Reflexive pointer	
Lexical key	object	box letter
object	0.83	0.78
box letter	0.79	0.85

Table 12: Mean principal cosine between the top 10 directions of the lexical and reflexive subspaces at layer 18 for each combination of key type

I Representational Similarity Analysis

We conduct a representational similarity analysis (RSA) on the identity codes. We take an 82×82 matrix of cosine similarity between the vectors for each representation and compare it to the same matrix calculated from the model's centered activations over the target token while reading an ordinary sentence, no retrieval involved. The matrices correlate at 0.53 for the lexical vectors and 0.60 for the reflexive, roughly twice that of the simple embeddings themselves (0.26 and 0.31 respectively), see Figure 14.

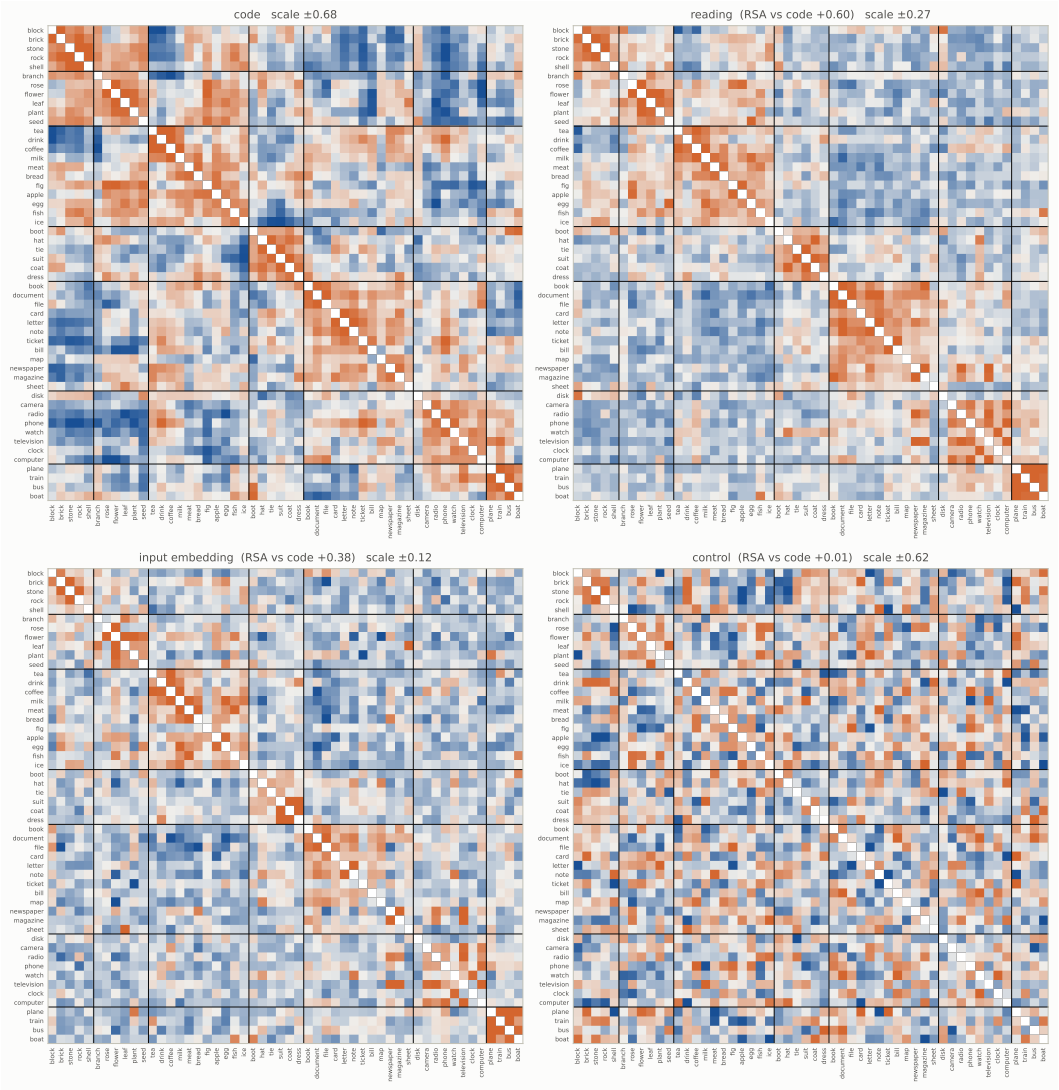


Figure 14: RSA of mechanism subspace vectors, centered residual stream vectors of representation in natural sentences, and embedding vectors.

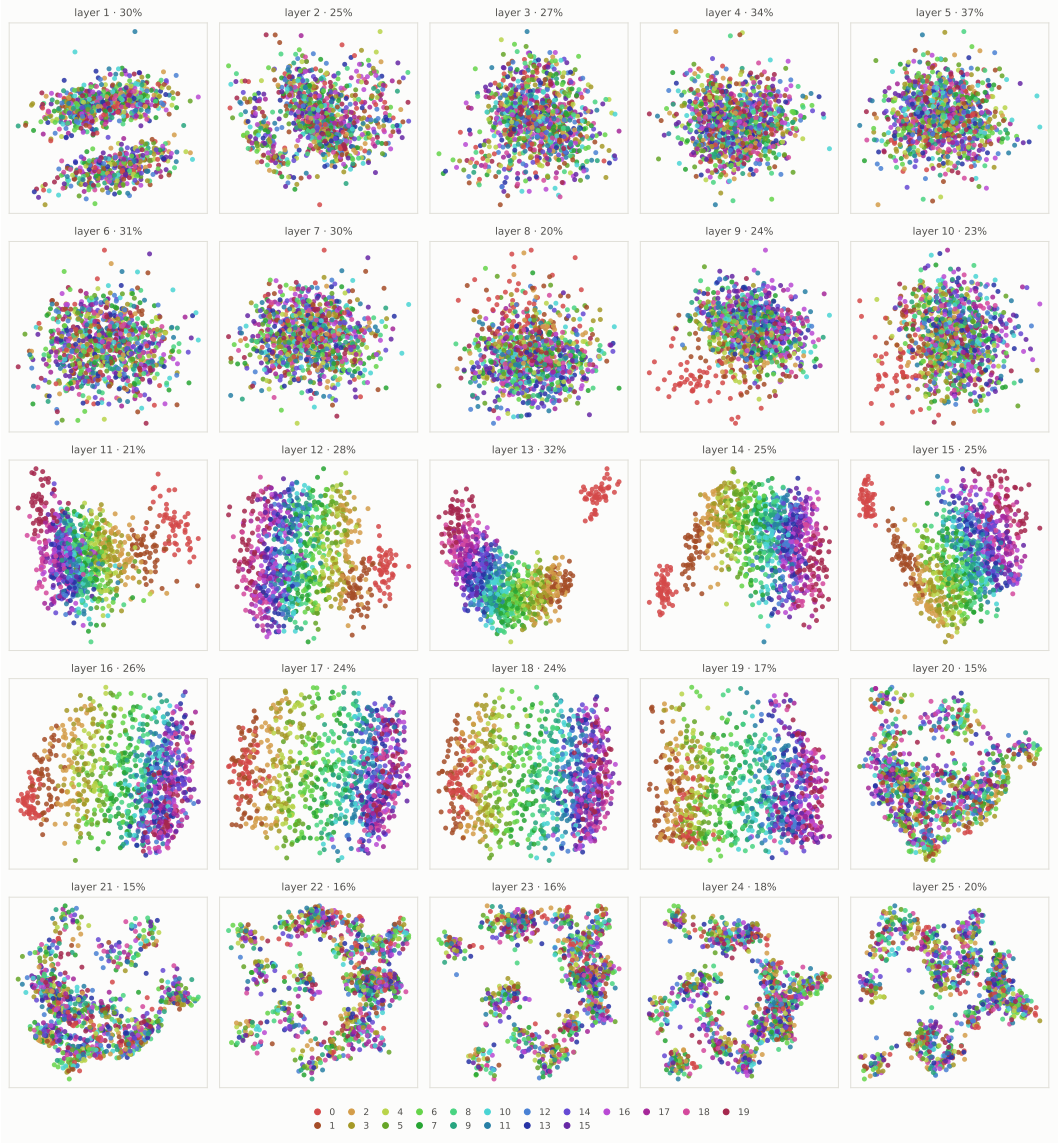


Figure 15: PCA of final token residual streams at all layers on the box to object task format.

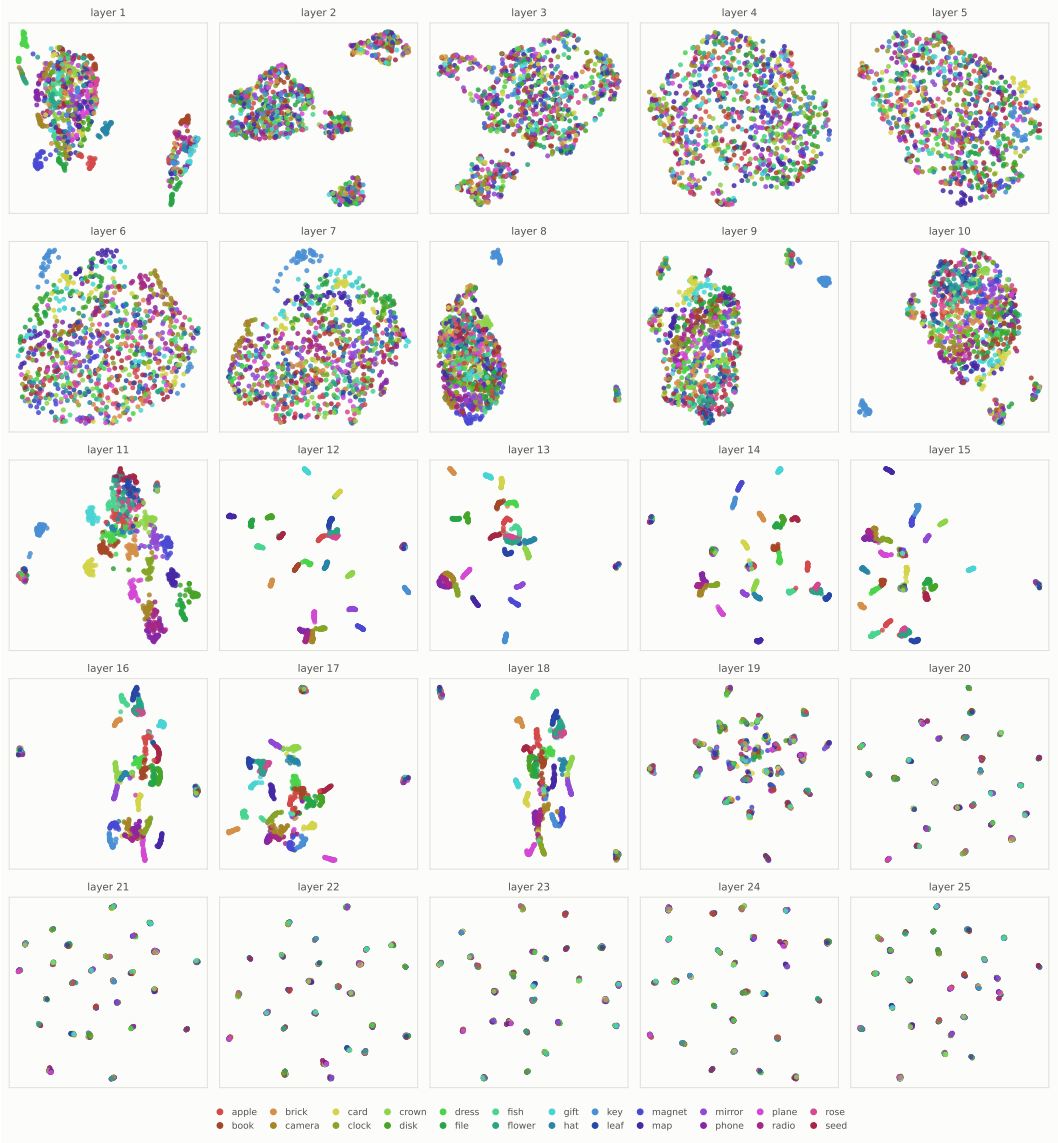


Figure 16: UMAP of final token residual streams all layers on the object to box task format.

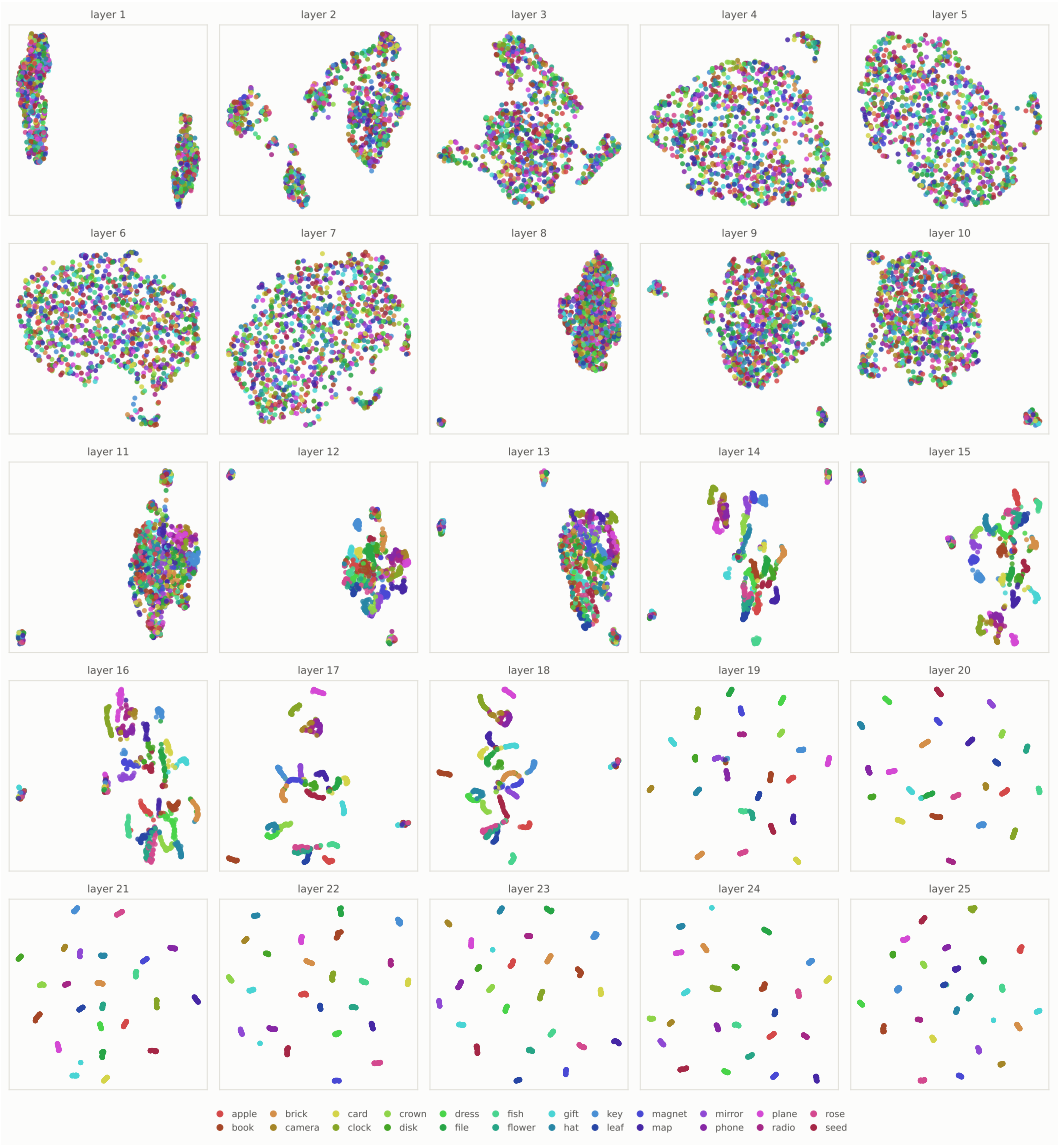


Figure 17: UMAP of final token residual streams at all layers on the box to object task format.