A retrieval conditioned rebinding circuit for dynamic entity tracking in large language models

Soyoung Oh [1] and Vera Demberg [1,2]

[1]Saarland University
[2]Max Planck Institute for Informatics
{soyoung, vera}@lst.uni-saarland.de

Abstract

To interpret context correctly and retrieve relevant information, large language models must bind entities to their attributes and update these bindings as state changes. We analyze how LLMs implement this binding process in a dynamic state tracking. Using causal interventions, we identify a *retrieval conditioned rebinding mechanism*, a compact attention head circuit that propagated binding information and when the entity is queried, uses the updated binding to retrieve the corresponding attribute. Across Gemma and Llama models, this circuit supports rebinding behavior, but the representational signature of the mechanism differs across model families. In Gemma models, the binding signature is clearly expressed in the query/key subspaces of the relevant attention heads, whereas in Llama models, the binding information is carried primarily in key vectors. Overall, our results reveal an interpretable mechanism for context dependent state tracking in LLMs[1].

1 Introduction

Accurately maintaining entity specific information over extended contexts is a fundamental requirement for a long context language understanding. Prior work has shown that large language models can succeed at entity tracking, yet their behavior remains inconsistent across settings (Kim and Schuster, 2023; Li et al., 2022; Kim et al., 2024). This raises a question about the nature of the underlying computation. Although recent studies have probed the mechanisms behind entity tracking, they have not fully examined settings that require explicit state transitions (Prakash et al., 2024) or naturalistic language contexts (Li et al., 2025). In this work, we study the mechanism for an entity tracking in a natural language setting with dynamic state updates.

[1]The code is publicly available at https://github.com/sori424/dynamicET.

We frame this problem through the lens of binding (Feng and Steinhardt, 2024; Feng et al., 2025), a representational capacity that supports compositional reasoning (Fodor and Pylyshyn, 1988). The binding refers to a process of associating the features of an object with the object itself, while keeping those features distinct from those of other objects (Treisman, 1996). Our interest is not only whether language models can form such associations, but also whether they can update them when the underlying state changes.

Consider a simple example involving three boxes and three objects, followed by a state update which is adapted from Kim and Schuster (2023):

Context: Box R ● contains the rabbit ●. Box S ▲ contains the sock ▲. Box T ■ contains the toy ■. Swap the items of Box S ▲ and Box R ●.
Question: Which item does **Box R** contain?

Answer: sock ▲ (1)

To answer this question correctly, a model must first represent the initial assignments `contains(R,rabbit)`, `contains(S,sock)`, and `contains(T,toy)`. Feng and Steinhardt (2024) showed that the model solves this *binding problem* by relying on binding IDs (●, ▲, ■), represented as vectors that are added to lexical information in the activation space. A natural hypothesis for how the model handles the state update is a **Global State Update**. Under this account, the model fully re-encodes the post update state immediately after the swap operation, analogous to mental simulation in cognitive accounts of language comprehension and reasoning (Johnson-Laird, 1983). On this

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 28229–28243
October 24-29, 2026 ©2026 Association for Computational Linguistics

view, the model maintains a situation model of the described world and incrementally updates it at the swap into a complete post swap configuration. For example, right after the `Swap` operation, the latent context would encode updated relations such as `contains(R,sock)` and `contains(S,rabbit)`. Answering then requires only reading off the object currently bound to the referenced box in this updated latent state.

An alternative hypothesis is the **Retrieval Conditioned Rebinding**. Under this account, the model does not re-encode the full situation immediately after each state transition. Instead, the model may preserve compact swap related information and apply it only during answer retrieval. Under the binding framework, therefore, a `Swap` operation need not permute all box–object bindings. When the final question asks about `Box R`, the model would use the locally updated binding ID for `Box R`, e.g., from ● to ▲, and then use ▲ to retrieve the associated object, *sock*. The answer is therefore obtained through a local rebinding followed by binding ID based readout, rather than by reconstructing a globally updated post swap state of the entire context.

We investigate this question using the dynamic state tracking task (Kim and Schuster, 2023) as in Example 1, together with causal interventions on LLM activations. The analysis proceeds in four steps and is run on four instruction-tuned models {Gemma2-9B, Gemma3-12B, Llama3.2-3B, and Llama3.1-8B} to test generalizability of the circuit:

1. We use causal mediation analysis (Prakash et al., 2025; Lieberum et al., 2023; Gur-Arieh et al., 2025) to localize the token level routes through which role specific information influences the final prediction. We observe that the target object is retrieved directly from its original contextual occurrence, whereas information about the boxes are mediated through multiple intermediate token positions. This dissociation supports a *retrieval conditioned rebinding*, where swap-relevant box information is transferred through intermediate positions and applied during readout time, rather than stored as a globally updated box–object state.

2. Next, we refine these coarse routes using path patching (Wang et al., 2023; Prakash et al., 2024). This reveals a compact attention head circuit, which consists of five functional groups, each associated with token positions identified in the previous step. The circuit, comprising 3–10% of the full model components, recovers near full model performance on the task.

3. To discern the role of each attention head, we conduct role specific interchange intervention (Vig et al., 2020; Prakash et al., 2025; Geiger et al., 2020). We find that one group functions as a pointer, which is sensitive to address selection, suggesting that these heads mediate binding identifier routing.

4. To test whether this binding identifier is causally involved in predicting the final answer, we adapt the binding intervention (Feng and Steinhardt, 2024) to the subspaces of the relevant attention head group. We observe that modifying the binding identifier redirects the model's attention to the corresponding object and shifts the final logits accordingly, but with different signatures across model families.

2 Task formulation and rebinding hypothesis

In the dynamic entity tracking task, LLMs represent an abstract binding ID shared by each box and its associated object, analogous to the binding mechanism proposed by Feng and Steinhardt (2024); Feng et al. (2025). Under this account, successful reasoning depends on preserving and updating box–object relations at the level of binding IDs, rather than at the level of surface token identity.

Let $\mathcal{B}$ denote a set of boxes and $\mathcal{O}$ a set of objects. A task instance of size n consists of distinct boxes $B_0, \ldots, B_{n-1} \in \mathcal{B}$ and distinct objects $O_0, \ldots, O_{n-1} \in \mathcal{O}$, arranged into an initial context $c_0 = \mathrm{ctxt}(B_0 \mapsto O_0,\ \ldots,\ B_{n-1} \mapsto O_{n-1})$. So, in Example 1, this corresponds to $c_0 = \mathrm{ctxt}(R \mapsto \mathit{rabbit},\ S \mapsto \mathit{sock},\ T \mapsto \mathit{toy})$.

Binding mechanism. We assume that each initial pair (B_k, O_k) is associated with a shared binding ID k. The box representation encodes the box identity together with this binding ID, and the object representation encodes the object identity together with the same ID. We denote these representations as $Z_{B_k} = \Gamma_B(B_k, k), \quad Z_{O_k} = \Gamma_O(O_k, k)$.

Under this view, answering a question about B_k requires retrieving the object whose binding ID matches the one assigned to B_k.

Rebinding mechanism. The swap operation $\mathrm{swap}(i, j)$ exchanges the objects associated with boxes B_i and B_j. A natural interpretation for this transition is a *global state update*, where the model constructs a complete post swap context right after the change. For example, applying $\mathrm{swap}(0, 1)$ in Example 1 yields updated context $c_1 = \mathrm{ctxt}(R \mapsto sock,\ S \mapsto rabbit,\ T \mapsto toy)$.

Under this account, the model answers by reading off the object bound to the referenced box in the updated latent state. On the other hand, the model may not construct the entire swapped world state in advance. Instead, it may preserves the original bindings and reinterpret them only when retrieval demands it. This motivates a more targeted alternative, the *retrieval conditioned rebinding* hypothesis. Under this account, the model preserves the original box–object representations from c_0 and applies the swap as a remapping over binding IDs during answer retrieval. We formalize this remapping as

$$\rho_{i,j}(k) = \begin{cases} j & \text{if } k = i, \\ i & \text{if } k = j, \\ k & \text{otherwise.} \end{cases}$$

Given a question about box B_k, the model applies the readout permutation $\rho_{i,j}$ to the referred binding ID, yielding $k' = \rho_{i,j}(k)$, and then retrieves the object representation whose stored binding ID matches k', $\Gamma_O(O_{k'}, k')$.

In Example 1, the referenced box is R, whose original binding ID is 0. After $\mathrm{swap}(0, 1)$, retrieval conditioned rebinding maps this ID to 1, allowing the model to retrieve $\Gamma_O(sock, 1)$ from the original context and return $sock$. To test these hypotheses, we start with causal mediation analysis to trace how information flows to the final answer position.

3 Tracing information flow of crucial input tokens

We focus on three functionally distinct token roles: the reference box (Box R), the swap target box (Box S), and the target object (sock). We conduct three separate intervention experiments: Exp 1 tests how information associated with the reference box flows by combining contextual content with the swap operation; Exp 2 tests how information from the swap target box is transferred to the question; Exp 3 reveals how the object value is first instantiated and then routed through to determine the answer.

Causal mediation analysis. To trace how answer-relevant information carried by role-bearing tokens flows into the final prediction, we conduct a coarse grained information flow analysis by using interchange interventions (Prakash et al., 2025; Geiger et al., 2021). For each role, we construct an original input x_{orig} and a counterfactual input x_{counter} that differ only in the role-bearing tokens relevant to that experiment. We then patch residual stream activations from x_{orig} into the corresponding token layer position of x_{counter}. For example, patching from an original context such as `Box S contains the sock` into a counterfactual context such as `Box S contains the cup` tests whether the patched activation is sufficient to shift the model from the counterfactual answer behavior, e.g., `cup`, toward the original expected answer, e.g., `sock`. We measure the interchange intervention accuracy (IIA), which indicates the extent to which patching that position causes the model to recover the original answer behavior (Geiger et al., 2022). A higher IIA means that position is a candidate for carrying causal signal for the answer prediction.

Results. Figure 1 shows the aggregated result of the causal mediation analysis (per-experiment results in Appendix Figure 6) over $N = 300$ examples. The effects indicate where information associated with each role contributes to the final answer prediction. Information associated with the reference box (Box R) and swap target box (Box S) does not appear to be routed from their initial contextual mentions. Instead, their causal effects emerge mainly after the `Swap` operation and propagate toward the final prediction position. This pattern suggests that the model does not answer by directly retrieving the swap target box's original contextual content. Consistent with this, patching the readout position (':') has a stronger causal effect than patching the swap operation position ('.' token in the swap sentence), suggesting that the final binding is resolved primarily during retrieval (Figure 7).

In contrast, information about the answer object (sock) is routed more directly. The object token's residual stream contributes to the final prediction in later layers, rather than being mediated through the box label tokens. This aligns with prior findings (Prakash et al., 2025, 2024; Lieberum et al., 2023; Gur-Arieh et al., 2025).

Together, this pattern contrasts with what we would expect under a *global state rebinding*. If

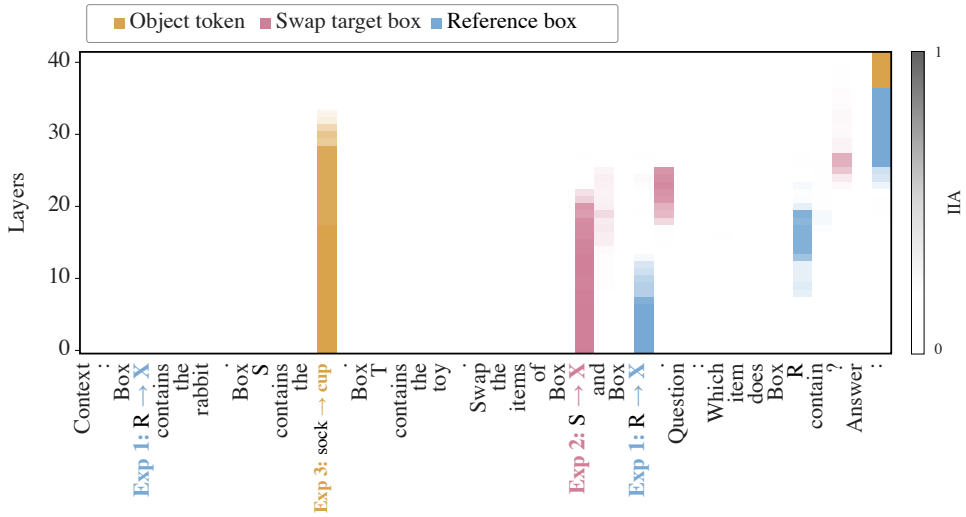


Figure 1: Information flow of crucial tokens using interchange interventions in Gemma-9B. **y-axis:** model layers; **x-axis:** shared original prompt template. Each colored token is varied in a separate experiment: Exp1. Reference box, Exp2. Swap target box, Exp 3. Object, all other black tokens are unchanged; **cell:** reports normalized IIA, darker cells indicating stronger causal mediation.

the model reconstructed a full post swap state, the answer object should be mediated through the updated box representation after the Swap. The fact that this is not the case suggests that the model does not rely primarily on a globally updated box–object representation. Rather, the observed pattern supports the *retrieval conditioned rebinding*, as the model appears to compute the relevant post swap box relation separately from object retrieval. That is, the Swap operation mediates a rebinding from the referenced box to its swapped target which then guides retrieval of the answer object from its original contextual position in later layers. This finding aligns with Tang et al. (2026), who argue that LMs do not incrementally track state updates across tokens but instead aggregate relevant information in parallel once the query becomes available. We further validate this finding in more complex settings by increasing the number of boxes and an additional distractor swap where the results remain consistent (Appendix A.1).

4 Localizing the rebinding mechanism

The causal mediation analysis provides token level evidence for a *retrieval conditioned rebinding* mechanism. To verify this mechanism at the attention head level, we next apply path patching (Wang et al., 2023; Prakash et al., 2024).

4.1 Identifying circuit components with path patching

Concretely, for each example ($N = 100$) we construct a clean input x_{orig} and a corrupted input x_{noise}, obtained by randomizing the box–object assignments while preserving the prompt structure. Let O_k be the correct answer for x_{orig}. We compute the clean probability $p_{\text{orig}} = p_M(O_k \mid x_{\text{orig}})$ and the patched probability $p_{\text{patch}} = p_{M_{\text{patch}}}(O_k \mid x_{\text{orig}})$, where the patched activation comes from x_{noise}. We add the paths with the most negative $(p_{\text{patch}} - p_{\text{orig}})/p_{\text{orig}}$ at each selection iteration.

We use two forms of path patching to distinguish two forms of sender–receiver interaction: *V-patching* and *Q-patching* (marked as V, Q in Figure 2). In V-patching, we replace the value side input to a receiver head to test whether it reads information written by an upstream sender at the attended source position. In Q-patching, we replace the query side input at the receiver head's target position to test whether an upstream sender shapes the receiver's query, thereby changing its attention pattern.

Rebinding circuit. We use the five functional roles employing Prakash et al. (2024) as an interpretive scaffold, not as a pre-specified circuit. These roles describe candidate rebinding subcomputations (Figure 2): (A) Answer retriever, (B) Dereferencer, (C) Position updater, (D) Swap position transmitter, and (E) Binding anchor. The heads in our circuit are selected independently through

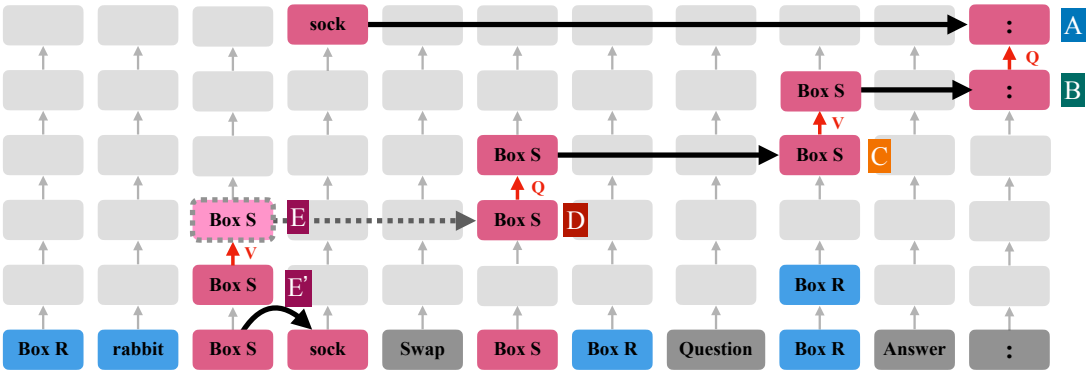


Figure 2: Path patching circuit for the retrieval conditioned rebinding mechanism. Rows denote functional head groups A–E at specific tokens; Columns denote layers. Horizontal arrows indicate attention mediated value routing. Vertical dashed arrows indicate residual stream composition between head groups, with V and Q denoting value and query mediated composition. Information flows via the chain (E $\xrightarrow{V}$) D $\xrightarrow{Q}$ C $\xrightarrow{V}$ B $\xrightarrow{Q}$ A $\rightarrow$ logit. E′ marks the context side box–object binding formation, while E marks a later binding anchor representation that writes the binding information into the readout circuit. The dotted Group E path is model dependent: it is active in some models, but not in Gemma-9B after pruning – binding anchor information may be routed by other heads or already available in the destination position's residual stream at readout.

causal pruning on our task and model. The rebinding circuit computation proceeds from Group E to Group A. Group E acts as a binding anchor at the context target box position, writing box identity information into the residual stream. Group D, the swap position transmitter, reads this information through V-composition and propagates it along the swap path. Group C, the position updater, receives information via Q-composition, which shapes Group C's attention toward the appropriate earlier box position to the updated binding. Group B, the dereferencer, then reads the resolved binding information through V-composition. Finally, Group A consists of answer retriever heads at the final position that representation supplied by Group B shapes Group A's attention over target object token, increasing the logit of the correct answer (i.e., sock). We used a five stage greedy path patching process. In each stage, all attention heads were scored as candidate senders over 100 prompts; the top-k heads with the most negative mean relative target probability change were retained. The fixed top-k values were 50,50,25,10, and 10 for Groups A-E respectively following the logic of the prior work (Prakash et al., 2024), and the procedure stopped after Group E with no early stopping.

Circuit evaluation. We evaluate the discovered rebinding circuit using candidate accuracy where a prediction is counted as correct if the gold object O_k receives the highest logit among the candidates.

Let F denote candidate accuracy averaged over $N = 300$ held-out examples, disjoint from the examples used to identify the circuit. We define $F(M)$ for the corresponding full model candidate accuracy and $F(Cir)$ as the circuit accuracy. We also set a random baseline by sampling 10 random subsets of attention heads with the same size as the candidate circuit and applying the same mean ablation procedure. We prune the candidate circuit using a contribution based criterion (Wang et al., 2023). For each head $v \in Cir$, and a subset of other circuit heads $K \subseteq Cir \setminus \{v\}$, we define the contribution of v as $(F(Cir \setminus K) - F(Cir \setminus (K \cup \{v\})))/F(Cir \setminus (K \cup \{v\}))$. Heads with contribution score less than 1% are pruned.

Results. We find that only 38 out of 672 attention heads (5.7%), are sufficient to recover most of the model's rebinding behavior. The circuit reaches 0.89 accuracy compared with 1.00 for the full model and 0.34 for a random head baseline, indicating that the relevant computation is localized in a compact attention head circuit. The selected heads and additional evaluation metrics are reported in Appendix B. Notably, the pruned circuit retains no heads from Group E. Since pruning removes attention head outputs but preserves the residual stream, binding relevant information may already be available in the relevant token positions due to earlier context side binding formation, marked as E′ in Figure 2, and then carried forward through the residual stream.

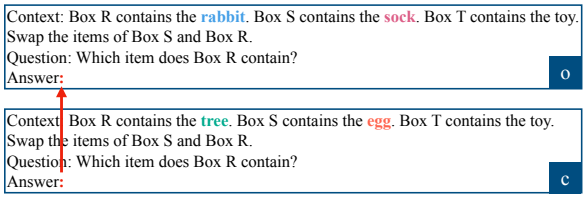
4.2 Identifying address retrieval routing head group

The previous path patching results suggest that at ':' position, the model first determines the relevant source position through attention routing, and then retrieves the object information stored at that position. This resembles the lookback mechanism of Prakash et al. (2025), where retrieval involves matching a later pointer token to an earlier address token, so that the model attends to the earlier token and reads out the associated payload. We therefore apply interchange interventions (Vig et al., 2020; Geiger et al., 2020) to the attention edges of each head group, testing whether its effect is carried by edges that select the retrieval address.

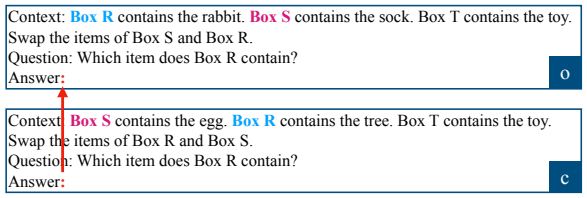
Patch construction and target logits. We construct paired original and counterfactual inputs ($N = 300$) and patch the attention patterns of the selected heads from the counterfactual run into the corresponding heads at the ':' position of the original run. We then measure the change in the logit of object token O_k, $\Delta\text{logit} = \text{logit}(O_k)_{\text{patched}} - \text{logit}(O_k)_{\text{original}}$. We use two variants. In the *content control intervention* (Figure 3a), the object tokens are changed while the box labels and swap structure are kept fixed. Since the retrieval address is unchanged, e.g., (Box R, 0) and (Box S, 1), patching only the attention pattern should continue to route retrieval to the same source position in the original prompt, rather than transferring the counterfactual object token. Thus, $\Delta\text{logit}(\texttt{egg})$ serves as a control for whether attention pattern patching alone transfers counterfactual object content independently of address selection.

In the *pointer intervention* (Figure 3b), we change the binding structure, (Box S, 0), (Box R, 1), so that the counterfactual prompt requires retrieval from a different source position. If a head group controls address selection, then patching its counterfactual attention pattern into the original run should shift the model toward the object located at the counterfactual source position in the original prompt (i.e., `rabbit`, the object stored at binding ID 0 in the original prompt).

Results. As shown in Figure 4, in the pointer intervention, patching the counterfactual attention pattern for Group B substantially increases $\Delta\text{logit}(\texttt{rabbit}) = 6.79$. In contrast, the same group produces only a small change in the content control target $\Delta\text{logit}(\texttt{egg}) = 0.26$. This dis-



(a) **Content control intervention:** If the heads mediate address routing rather than object content transfer, attention pattern patching should not increase the counterfactual object logit, so we expect $\Delta\text{logit}(\texttt{egg}) \approx 0$.



(b) **Pointer intervention:** If the heads mediate retrieval addressing, we expect $\Delta\text{logit}(\texttt{rabbit}) > 0$, where `rabbit` is the token at the counterfactual position 0 in the original prompt.

Figure 3: Attention pattern interchange paired prompts: o (original), c (counterfactual).

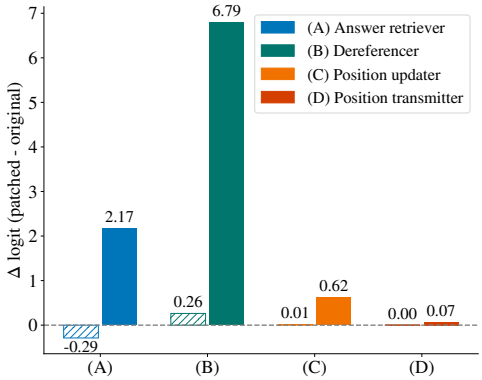


Figure 4: Role specific head patching effects for Gemma-9B. ■ $\Delta\text{logit}(\texttt{rabbit})$ is represented by fully colored bars, while ▨ $\Delta\text{logit}(\texttt{egg})$ is represented by diagonally hatched bars.

sociation indicates that Group B primarily mediates retrieval routing through address selection, rather than transferring counterfactual object content. Group A, the answer retriever heads, also increases the pointer target logit ($\Delta\text{logit}(\texttt{rabbit}) = 2.17$), while decreasing the content control target logit ($\Delta\text{logit}(\texttt{egg}) = -0.29$). This suggests that Group A may also participate in routing to the retrieved source position. Overall, the results identify Group B as the clearest address mediated retrieval routing group, with weaker contributions from Groups A and C[2].

[2]We exclude Group E because no heads satisfy the pruning criterion.

5 Probing binding ID matching in Q/K subspaces of dereferencer heads

The attention pattern interchange results show that Group B plays a causal role in redirecting retrieval across binding positions. Therefore, we ask whether Group B's pointer role arises from a binding ID comparison in its Q/K subspaces. Since a head's attention scores are computed by query–key dot products, Group B could implement retrieval routing by binding ID k at ':' position, while keys at earlier context tokens encode corresponding binding ID k. Matching binding IDs would then cause the head to attend to the corresponding answer object token. We test this with binding ID interventions in Q/K space of Group B heads.

Following the additive representation hypothesis (Elhage et al., 2021) and the binding ID intervention framework (Feng and Steinhardt, 2024), we treat binding IDs as approximately linearly represented directions in the activation space. We estimate a binding ID direction in the query and key spaces of the selected Group B heads. For a head h in layer ℓ, let

$$A_{\ell,h}(t,s) = \mathrm{softmax}\left(\frac{q_{\ell,h}(t)^\top k_{\ell,h}(s)}{\sqrt{d_h}}\right)$$

denote the attention pattern from a target position t to a source position s. In our task, the target position $t_:$ is the ':' token, and the source positions are the context tokens corresponding to box and object tokens.

Binding ID directions in Q/K subspace. We estimate binding ID directions by averaging Group B query and key vectors over $N = 100$ prompts with randomized lexical items, so that lexical variation is averaged out. Let $r(x)$ denote the retrieval binding ID in prompt x. For each binding ID k, we compute the mean query vector at the ':' position as $\mu^Q_{\ell,h}(k) = \mathbb{E}_{x:\, r(x)=k}\left[q^x_{\ell,h}(t_:)\right]$. For the key side, let $S_k(x)$ denote the set of context box/object token positions carrying binding ID k in prompt x, then $\mu^K_{\ell,h}(k) = \mathbb{E}_x\left[\frac{1}{|S_k(x)|}\sum_{s\in S_k(x)} k^x_{\ell,h}(s)\right]$.

Q/K intervention. We define shift directions as differences between the mean Q/K representations for two binding IDs:

$$\Delta^Q_{\ell,h}(i\to j) = \mu^Q_{\ell,h}(j) - \mu^Q_{\ell,h}(i),$$
$$\Delta^K_{\ell,h}(i\to j) = \mu^K_{\ell,h}(j) - \mu^K_{\ell,h}(i).$$

Assuming that binding ID is approximately linearly represented, $\mu(j) - \mu(i)$ estimates the representational shift from ID i to ID j. We use these directions for causal interventions in the query and key subspaces. For each original prompt requiring retrieval from binding ID i, we pair a counterfactual binding ID j and apply three interventions.

In the *query-side intervention*, we shift the query vector at the ':' position from ID (i) to (j) by

$$q_{\ell,h}(t_:) \leftarrow q_{\ell,h}(t_:) + \Delta^Q_{\ell,h}(i\to j),$$

This tests whether changing the binding ID requested at the answer position redirects Group B attention from source positions carrying ID i to j.

In the *key-side intervention*, we keep the query fixed but change the binding IDs represented by the context tokens at box only, object only, and both positions. Let S^r_k denote the selected context positions associated binding ID k, where $r \in \{\mathrm{box}, \mathrm{object}, \mathrm{both}\}$. We then apply

$$k_{\ell,h}(s) \leftarrow k_{\ell,h}(s) + \Delta^K_{\ell,h}(j\to i),\ s\in S^r_j,$$
$$k_{\ell,h}(s) \leftarrow k_{\ell,h}(s) + \Delta^K_{\ell,h}(i\to j),\ s\in S^r_i.$$

If Group B attends by Q/K binding ID matching, the unchanged ID i query should now match the shifted keys at S^r_j, redirecting attention to the counterfactual context. The three variants test whether this match is carried by keys of box, object, or both.

In the *Q+K intervention*, we apply both interventions together. The query is shifted from ID i toward j, while the keys at the original source positions S_i are also shifted from ID i toward j. Thus, if Group B computes the pointer through Q/K compatibility, the shifted query should again match the original source positions S_i, now via their shifted keys. Thus, the combined intervention should cancel the redirection observed in the Q-only and K-only interventions.

Metrics. We report three metrics averaged over $N = 300$ examples: (i) ΔR measures how much the intervention shifts attention from context tokens associated with the original binding ID i toward counterfactual binding ID j; positive values indicate a shift toward the counterfactual binding. (ii) Δlogit measures the corresponding change in the final output logit margin favoring the counterfactual answer over the original answer. (iii) Switch fraction measures how often the intervention changes the dominant attended binding ID from i to j. Formal definitions of these metrics are provided in Appendix D.

Condition	ΔR	Δlogit	Switch
Random	-0.28	0.48	0.06
Q int. (↑)	**2.66**	**4.25**	**0.47**
K int. (box+obj) (↑)	**2.08**	**3.51**	**0.31**
K int. (box) (↑)	1.92	0.04	0.42
K int. (obj) (↑)	2.25	4.35	0.35
Q+K int. (box+obj) (↓)	**-0.12**	**1.57**	**0.12**

Table 1: Q/K binding-ID interventions on Group B in Gemma-9B. The random control uses a matched norm random direction. Gray cells mark the diagnostic interventions.

Results. The Q/K binding interventions indicate that Gemma-9B selects addresses via binding ID matching (Table 1). Query side intervention strongly redirects attention toward the counterfactual binding ID ($\Delta R = 2.66$), with corresponding behavioral shifts ($\Delta\text{logit} = 4.25$, switch fraction 0.47). Key side interventions also redirect attention, especially when both box and object keys are shifted ($\Delta R = 2.08$), while object only key shifts produce the largest logit change ($\Delta\text{logit} = 4.35$). In contrast, box only key shifts affect attention but not logits, suggesting that box tokens support address selection whereas object tokens are more directly coupled to readout. Crucially, the combined Q+K intervention cancels the effect ($\Delta R = -0.12$, switch fraction 0.12), close to the random direction control. Overall, Gemma-9B appears to implement address selection through Q/K binding ID matching, with object side key vectors most directly driving downstream answer readout.

6 Generalization of local state rebinding across models

We next examine whether retrieval conditioned rebinding reflects a general mechanism across Gemma-12B, Llama-3B, Llama-8B. We first use path patching to identify candidate rebinding circuit at the attention head level (Section 4.1). Then, we localize the pointer like heads that route binding information (Section 4.2) and perform binding ID interventions on these head groups (Section 5).

Rebinding circuit evaluation. Table 2 shows that the pruned rebinding circuits recover much of the full model behavior across models. The results with additional evaluation metrics are reported in Appendix Table 4.

Model	$F(M)$	$F(Cir)$	$F(Random)$	# Component
Gemma-9B	1	0.89	0.34	38 / 672
Gemma-12B	1	0.91	0.38	83 / 768
Llama-3B	0.71	0.71	0.34	42 / 672
Llama-8B	0.99	0.84	0.38	40 / 1024

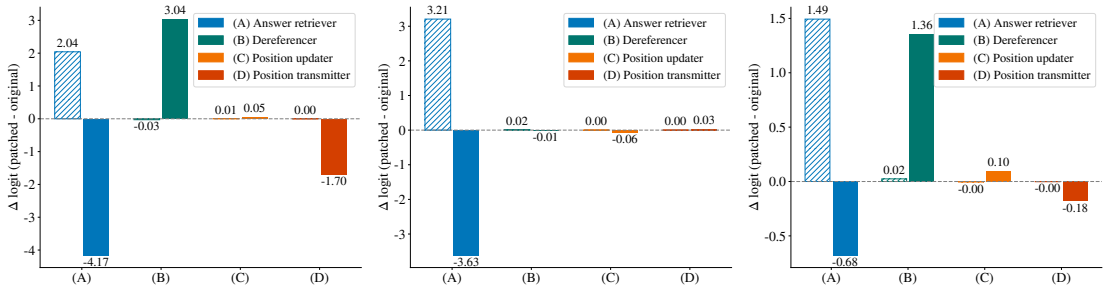
Table 2: Full model and rebinding circuit accuracy across Gemma-{9B, 12B}, Llama-{3B, 8B} with random baseline with number of attention heads retained. Additional experiments restricted to examples correctly answered by Llama-3B are presented in Appendix E.

Functional roles of routing heads. We next ask whether the head groups identified in the rebinding circuit play the same functional roles across models. As in Figure 5, across models (except Llama-3B), Group B remains important for address or pointer routing, where interchanging this group increases the $\Delta\text{logit}(\texttt{rabbit})$, while leaving $\Delta\text{logit}(\texttt{egg})$ comparatively unchanged. In contrast, Group A plays a larger role in answer object retrieval in the other models, where interchange on Group A increase the $\Delta\text{logit}(\texttt{egg})$ while decreasing the $\Delta\text{logit}(\texttt{rabbit})$. These results suggest a similar functional decomposition across models, with Group B primarily supporting address like routing and Group A contributing more directly to payload or answer retrieval.

Binding ID matching Q/K intervention. The intervention results show that the binding ID mechanism generalizes across model families, but with different Q/K-level implementations. As shown in Table 3, Gemma-12B preserves the weak Q/K matching pattern observed in Gemma-9B (query side intervention produces the strongest attention and logit shifts ($\Delta R = 0.26$, $\Delta_{\text{logit}} = 0.41$), while Q+K intervention cancels the effect ($\Delta R = -0.04$, $\Delta_{\text{logit}} = -0.49$)). In contrast, Llama models are more key dominated. That is, key side interventions produce the largest attention shifts, query side effects are weak, and Q+K intervention remains positive ($\Delta R = 1.69$ for Llama-3B; $\Delta R = 0.98$ for Llama-8B). Thus, local state rebinding appears across architectures, but Gemma models exhibit clearer Q/K matching whereas Llama models rely more on key side binding information.

7 Conclusion

We investigate how LLMs perform dynamic entity tracking in natural language contexts. Our findings support a mechanistic account in which LLMs do not simply reconstruct a complete updated world



(a) Gemma-12B (b) Llama-3B (c) Llama-8B

Figure 5: Role specific head patching effects across other models. ■ $\Delta\text{logit}(\texttt{rabbit})$ is represented by fully colored bars, while ▨ $\Delta\text{logit}(\texttt{egg})$ is represented by diagonally hatched bars.

Model	Condition	ΔR	Δlogit	Switch
Gemma-12B	Random	-0.12	-0.69	0.08
	Q int. (↑)	**0.26**	**0.41**	**0.04**
	K int. (box+obj) (↑)	**0.29**	**0.10**	**0.14**
	K int. (box) (↑)	-0.33	-0.05	0.12
	K int. (obj) (↑)	0.87	0.23	0.18
	Q+K int. (box+obj) (↓)	**-0.04**	**-0.49**	**0.08**
Llama-3B	Random	-0.10	0.21	0.05
	Q int. (↑)	**-0.46**	**0.15**	**0.01**
	K int. (box+obj) (↑)	**1.32**	**1.03**	**0.30**
	K int. (box) (↑)	2.23	0.06	0.37
	K int. (obj) (↑)	0.29	0.69	0.15
	Q+K int. (box+obj) (↓)	**1.69**	**0.95**	**0.34**
Llama-8B	Random	0.17	1.02	0.12
	Q int. (↑)	**0.18**	**0.83**	**0.06**
	K int. (box+obj) (↑)	**1.20**	**2.92**	**0.36**
	K int. (box) (↑)	1.59	-0.02	0.40
	K int. (obj) (↑)	0.78	2.70	0.26
	Q+K int. (box+obj) (↓)	**0.98**	**1.05**	**0.27**

Table 3: Q/K binding ID interventions on Group B. The random control uses a matched norm random direction. ΔR: change in log attention ratio toward the counterfactual over the original target; Δlogit: corresponding change in the counterfactual original object logit difference; Switch: switch rate of top attended context token from i to j. Gray cells mark the diagnostic Q-side, K-side and Q+K cancellation interventions.

state after each state changing operation. Instead, they combine abstract binding representations at retrieval time and perform pointer updates. That is, the model preserves the original object representations and uses swap related information to redirect retrieval to the appropriate binding ID at retrieval time. This mechanism provides a concrete circuit level explanation for context dependent retrieval in transformers.

At the same time, the mechanism is not entirely identical across models. Although Gemma and Llama models show evidence for retrieval conditioned rebinding, they differ in functional roles of the attention head groups and Q/K signatures associated with the relevant heads. This suggests that the rebinding mechanism may be a shared computational strategy, but its circuit level implementation varies across model families. Understanding these cross-model differences is an important direction for future work, particularly for clarifying which aspects of rebinding reflect model architecture, scale, or properties of the pretraining distribution.

Limitations

While our task setting enables precise causal intervention, it is highly controlled and may not capture the full range of mechanisms used in natural discourse. Entities are represented as alphabetically named boxes, objects are single tokens, and state changes are limited to a single swap operation. Moreover, although the discovered circuits preserve much of the full model performance, they may omit redundant or backup components. Future work should test whether the same circuit structure generalizes to multiple swaps, larger entity sets, other state transitions such as put and remove, and more naturalistic narratives.

Additionally, the binding ID directions used in the Q/K intervention experiments are estimated with mean difference probes under an approximate linear representation assumption. If binding information is represented nonlinearly, distributed across multiple subspaces, or dynamically transformed across layers, these interventions may underestimate the relevant mechanism.

Ethical Considerations

All experiments were conducted on a single NVIDIA H100 PCIe GPU with 80 GB of memory with synthetic prompts involving boxes, objects, and swap operations. This work therefore poses

minimal risks of data privacy and human subjects, although its GPU based analyses can incur the environmental costs. Specifically, the most computationally intensive experiment was path patching with Gemma-12B required ∼16 hours to complete. The intervention methods are also potentially dual-use, as they can be used both to diagnose and to manipulate model behavior.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC Starting Grant "Individualized Interaction in Discourse", grant agreement No. 948878). We gratefully acknowledge the stimulating research environment of the GRK 2853/1 "Neuroexplicit Models of Language, Vision, and Action", funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under project number 471607914.

References

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*.

Jiahai Feng, Stuart Russell, and Jacob Steinhardt. 2025. Monitoring latent world states in language models with propositional probes. In *The Thirteenth International Conference on Learning Representations*.

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

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

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. *Advances in neural information processing systems*, 34:9574–9586.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the third blackboxnlp workshop on analyzing and interpreting neural networks for NLP*, pages 163–173.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. 2022. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, pages 7324–7338. PMLR.

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

Philip Nicholas Johnson-Laird. 1983. *Mental models: Towards a cognitive science of language, inference, and consciousness*. 6. Harvard University Press.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855.

Najoung Kim, Sebastian Schuster, and Shubham Toshniwal. 2024. Code pretraining improves entity tracking abilities of language models. *arXiv preprint arXiv:2405.21068*.

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

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2022. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*.

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

Nikhil Prakash, Natalie Shapira, Arnab Sen Sharma, Christoph Riedl, Yonatan Belinkov, Tamar Rott Shaham, David Bau, and Atticus Geiger. 2025. Language models use lookbacks to track beliefs. In *The Fourteenth International Conference on Learning Representations*.

Zilu Tang, Qiao Zhao, Gabriel Franco, Derry Wijaya, Aaron Mueller, Sebastian Schuster, and Najoung Kim. 2026. Do language models track entities across state changes? *arXiv preprint arXiv:2605.30233*.

Anne Treisman. 1996. The binding problem. *Current opinion in neurobiology*, 6(2):171–178.

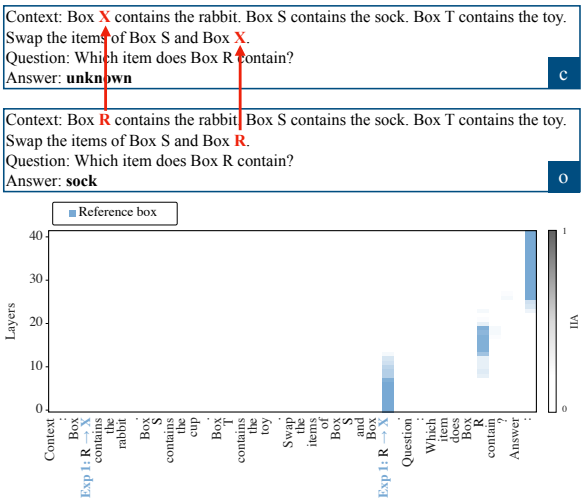
Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401.

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*.

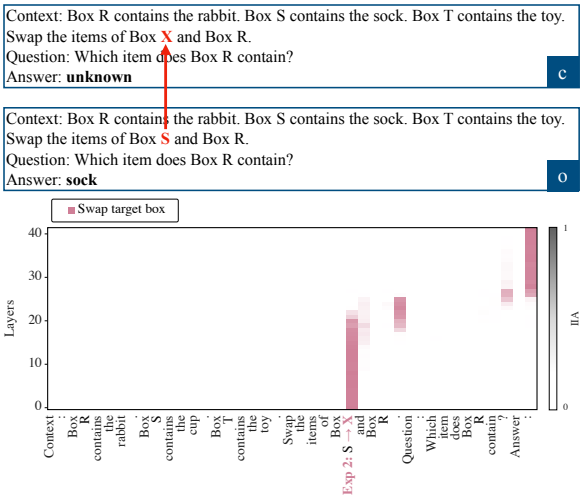
A Causal mediation analysis for tracing information flow

Figure 6 shows the causal mediation analysis setup and per-experiment results.

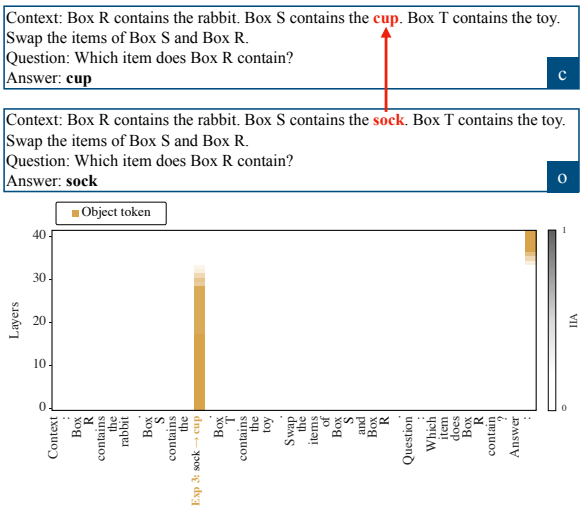
We additionally conduct experiment for patching the '.' position after the swap operation (post swap prefix) and compare it with when patching ':' position at the retrieval. If the post swap prefix already encodes the updated global state, then patching activations at post swap prefix positions should have an effect comparable to patching the activation at the final answer position. If rebinding is computed during retrieval, answer position patches should produce much stronger counterfactual transfer than prefix patches. We further repeat the same patch but questions across all boxes (i.e., Which item does Box {b} contain?, $b \in \{R, S, T\}$). If the same prefix patch transfers the whole counterfactual mapping for all three boxes (i.e., increase the logit for counterfactual mapping), then it indicates for global state update right after the swap operation. If it only helps a specific referenced box, then it is probably encoding the swap operation rather than a complete updated state. As in Figure 7, patching the post swap prefix produces only a localized counterfactual effect on the referenced box, whereas patching the answer colon position yields a stronger and more consistent effect across all the boxes. This pattern suggests that swap related information is present before the question, but the relevant binding is resolved mainly during retrieval rather than stored as a complete post swap state.



(a) Exp1. Reference box tokens.



(b) Exp2. Swap target box token.



(c) Exp3. Object token.

Figure 6: Information flow via causal mediation analysis per-experiment setup and result.

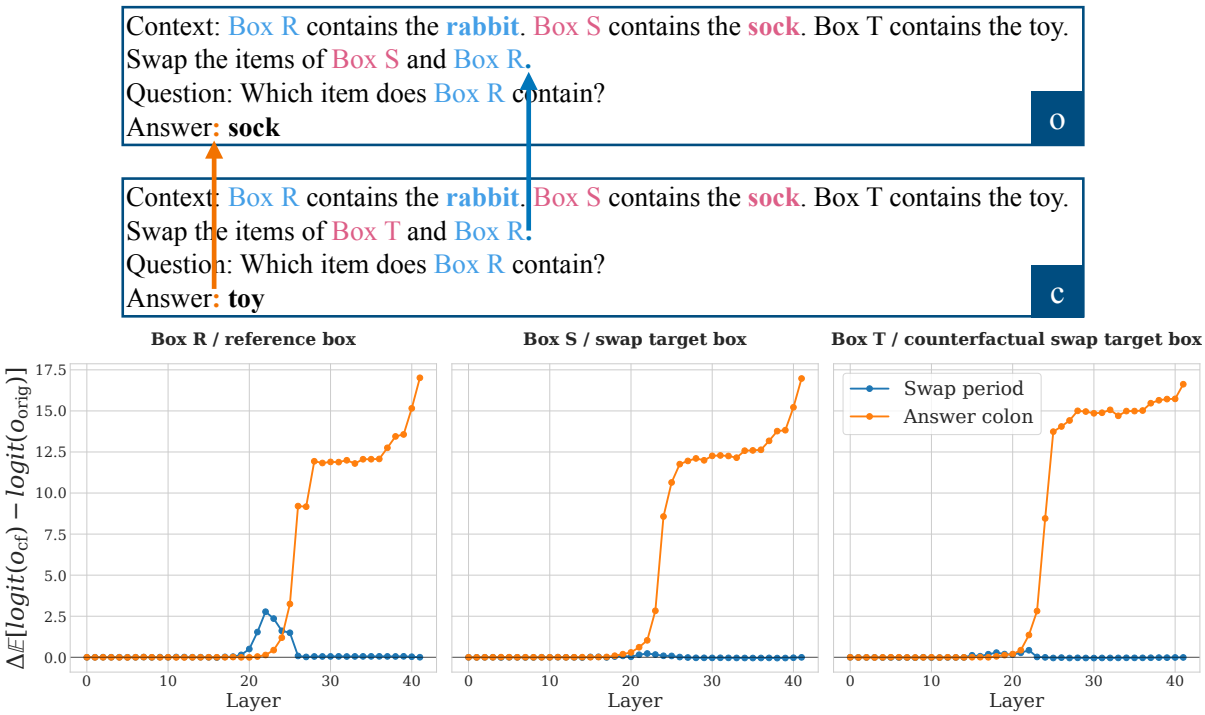


Figure 7: Additional causal mediation diagnostic comparing post swap prefix patching with readout position patching. **Top:** We patch activations from c into o either at the period following the swap instruction (post swap prefix patching; blue arrow) or at the answer colon position (readout patching; orange arrow). **Bottom:** Change in mean counterfactual minus original logit margin relative to the clean baseline. Positive values indicate that the patch shifts the model toward the counterfactual answer.

A.1 Generalization beyond the three-box setting

We extended these analyses to a setting with more boxes and an additional distractor swap, as shown in Figure 8 and Figure 9.

B Selected attention heads in rebinding circuit

Table 4 shows the mean candidate margin and mean label logit of the full model, pruned circuit, and random baseline. Table 6 reports the attention heads selected after pruning for each group in the rebinding circuit.

C Functional roles of attention heads groups

Appendix Figure 5 provides the full per-model results for the head role interventions as described in Section 6.

D Binding ID intervention metrics

For each example, let i denote the original binding ID and j denote the counterfactual binding ID induced by the intervention. Let S_k be the set of

	Mean candidate margin			Mean label logit		
Model	M	Cir	Rand.	M	Cir	Rand.
Gemma-9B	8.87	2.76	-0.57	20.33	12.35	7.67
Gemma-12B	17.83	14.10	-0.78	31.73	21.69	6.22
Llama-3B	0.84	2.32	-0.80	20.24	16.36	11.49
Llama-8B	4.72	3.70	-0.52	20.77	14.72	9.59

Table 4: Additional circuit evaluation metrics across models. We report candidate margin and mean label logit for the full model (M), the pruned rebinding circuit (Cir), and random head subsets of matched size (Rand.) on held out examples ($N = 300$).

context token positions associated with binding ID k, and let $A_{\ell,h}(t_:, s)$ denote the attention weight from the relevant query position $t_:$ to source position s for head (ℓ, h). We define the attention mass assigned to binding ID k as

$$M_k = \sum_{s \in S_k} A_{\ell,h}(t_:, s).$$

The change in log attention ratio is

$$\Delta R = [\log(M_j + \epsilon) - \log(M_i + \epsilon)]_{\text{patched}} - [\log(M_j + \epsilon) - \log(M_i + \epsilon)]_{\text{original}},$$

where ϵ is a small constant for numerical stability. A positive ΔR indicates that the intervention shifts attention from the original binding ID i toward the counterfactual binding ID j.

Let $z(O)$ denote the final output logit assigned to object token O, and let O_i and O_j be the objects associated with the original and counterfactual binding IDs, respectively. We define the change in final logit margin as

$$\Delta\text{logit} = [z(O_j) - z(O_i)]_{\text{patched}} - [z(O_j) - z(O_i)]_{\text{original}} .$$

Finally, the switch fraction measures the fraction of examples for which the dominant attended binding ID changes from i in the original run to j in the patched run. Let $o_n = \arg\max_k M^{(n)}_{k,\text{original}}, p_n = \arg\max_k M^{(n)}_{k,\text{patched}}$, then

$$\text{Switch} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\{o_n = i,\ p_n = j\}.$$

E Binding ID intervention on Llama-3B only on correct answered examples

Llama-3B results should be interpreted cautiously because it might be already weak on the task. However, we would not describe the found circuit accuracy as floor effect, where the circuit accuracy remains above the random head baseline (0.71; random: 0.34). We evaluated on the held-out dataset than the dataset that we found the circuit from to make sure that the found circuit is not just a random artifact. Moreover, we argue that the positive Q+K intervention for the Llama-3B model shows that Gemma model's binding Q/K mechanism does not hold for Llama model. We additionally conduct further experiment to address this point more directly.

We conduct additional head specific binding ID intervention only on the examples that the model answer correctly to get rid of the effect of the confounding error. As in Table 5, the model performs the similar as the original performance for the binding intervention where it uses all the examples. Therefore, the positive Q+K effect is probably not due to low task accuracy or failures. It does reflect the key-dominated mechanism.

	Q int.		K int.		Q+K int.	
	ΔR	Δlogit	ΔR	Δlogit	ΔR	Δlogit
All	-0.46	0.15	1.32	1.03	1.69	0.95
Only correct	-0.48	0.11	1.36	1.05	1.73	1.05

Table 5: Llama-3B intervention results for all vs. only corrected examples.

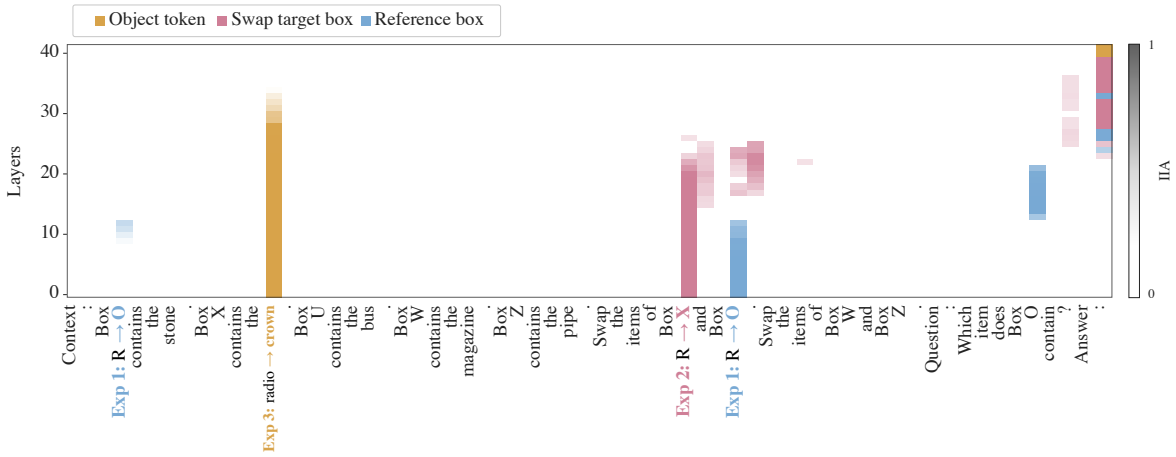


Figure 8: Information flow of crucial tokens using interchange interventions in Gemma-9B with larger number of boxes (N=5) and an additional swap distractor. **y-axis:** model layers; **x-axis:** shared original prompt template. Each colored token is varied in a separate experiment: Exp1. Reference box, Exp2. Swap target box, Exp 3. Object, all other black tokens are unchanged; **cell:** reports normalized IIA, darker cells indicating stronger causal mediation.

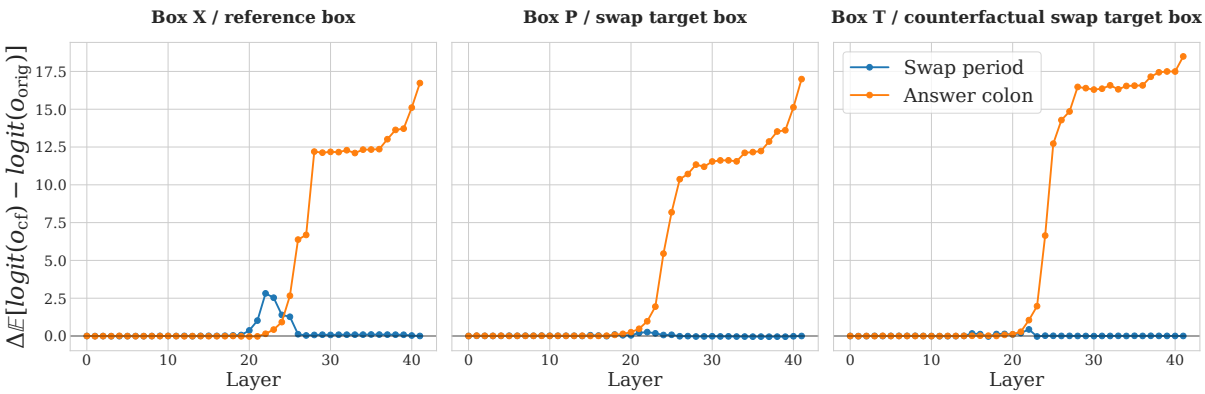


Figure 9: Additional causal mediation diagnostic comparing post swap prefix patching with readout position patching with 5 boxes and a distractor swap operation. Change in mean counterfactual minus original logit margin relative to the clean baseline. Positive values indicate that the patch shifts the model toward the counterfactual answer.

Model	Category	Attention heads (h, ℓ)
Gemma-9B	(A) Answer retriever	(15, 27), (7, 25), (0, 26), (1, 25), (8, 39), (1, 39), (11, 40), (7, 21), (13, 27), (12, 38), (10, 25)
	(B) Dereferencer	(12, 28), (4, 28), (1, 30), (5, 22), (14, 24), (3, 21), (8, 28), (10, 28), (6, 29), (12, 21), (12, 26), (7, 32), (15, 30), (4, 26), (4, 22), (15, 11), (2, 28)
	(C) Position updater	(3, 18), (7, 21), (3, 19), (14, 19), (10, 16), (1, 18), (2, 0), (11, 22), (13, 8)
	(D) Swap position transmitter	(0, 17)
	(E) Binding anchor	-
Gemma-12B	(A) Answer retriever	(2, 47), (1, 46), (14, 47), (4, 47), (10, 32), (7, 47), (4, 31), (13, 32), (2, 30), (0, 15), (15, 30), (4, 10), (8, 16), (12, 29), (5, 29), (11, 14), (3, 37), (1, 4), (13, 43)
	(B) Dereferencer	(6, 28), (0, 46), (11, 32), (2, 44), (11, 41), (14, 1), (13, 6), (3, 18), (14, 30), (8, 46), (10, 6), (4, 26), (1, 9), (4, 39), (12, 26), (10, 15), (9, 6), (3, 0), (7, 45), (9, 46), (7, 26), (3, 41), (12, 44), (2, 46), (1, 35), (10, 28), (3, 30), (10, 3), (11, 46)
	(C) Position updater	(6, 5), (8, 40), (2, 30), (0, 5), (14, 24), (4, 18), (6, 41), (9, 33), (2, 27), (15, 32), (14, 9), (14, 15), (13, 34), (8, 36), (9, 11), (10, 6), (14, 12), (11, 27)
	(D) Swap position transmitter	(14, 7), (3, 10), (7, 19), (9, 9), (2, 35), (1, 19), (6, 13), (10, 5), (6, 28)
	(E) Binding anchor	(8, 20), (13, 9), (7, 13), (10, 10), (11, 19), (6, 22), (6, 25), (4, 23)
Llama-3B	(A) Answer retriever	(5, 24), (9, 18), (0, 27), (2, 15), (17, 21), (6, 16), (22, 15), (1, 27), (20, 21), (0, 17), (17, 16), (9, 23), (11, 22)
	(B) Dereferencer	(15, 13), (2, 12), (1, 12), (5, 11), (23, 13), (17, 13), (18, 15), (3, 12), (11, 12), (16, 14), (14, 12), (19, 14), (10, 18), (6, 9), (3, 7), (7, 19), (16, 10), (7, 7), (2, 14), (1, 7)
	(C) Position updater	(11, 10), (9, 11), (23, 9), (1, 11), (20, 9)
	(D) Swap position transmitter	(22, 8), (19, 2), (7, 10), (2, 7)
	(E) Binding anchor	-
Llama-8B	(A) Answer retriever	(7, 27), (29, 22), (28, 16), (20, 18), (26, 17), (26, 16), (4, 17), (13, 24)
	(B) Dereferencer	(6, 14), (13, 13), (10, 21), (11, 15), (1, 16), (13, 20), (8, 15), (5, 14), (24, 17), (15, 13), (21, 13), (28, 18), (3, 26), (16, 16), (5, 17), (3, 16), (19, 22), (25, 12), (9, 15), (11, 22), (25, 16), (26, 11), (13, 26), (2, 13), (26, 24)
	(C) Position updater	(2, 10), (25, 12), (3, 10), (21, 13)
	(D) Swap position transmitter	(13, 8), (0, 10)
	(E) Binding anchor	(21, 5)

Table 6: Attention heads remaining after pruning across four models. Heads are reported as (h, ℓ), where h denotes the head index and ℓ denotes the layer index.