Language models can associate objects with their features without forming integrated representations

Simon Jerome Han
Department of Psychology
Stanford University
Stanford, CA 94305

James L. McClelland
Department of Psychology
Stanford University
Stanford, CA 94305

Abstract

Language models (LMs) are adept at using in-context learning to associate objects with their features – for example, given an input such as 'the table is small and orange and the phone is green and big, so the color of the phone is', they can correctly infer that the next word should be 'green'. How? One possibility is that they rely on integrated representations of objects that jointly encode feature information within specific token positions. Another is that they access disparate sets of feature-specific representations distributed across positions as needed. By applying causal mediation analysis on an LM performing a multi-object multi-feature association task, we find a small set of upper-layer attention heads that search for and copy feature-specific representations based on the demands of a specific query. These heads are sufficient and necessary for our task, suggesting that LMs do not rely on integrated object representations to complete it.

How do intelligent systems represent objects and their features? For humans, a dominant hypothesis is that perceiving an object (e.g. 'a phone') and its features (e.g. 'big', 'green') involves the rapid formation of a single integrated object representation (i.e. 'a big green phone') [27, 8]. This is thought to occur during both visual perception [28] and language processing [11], and is so deeply embedded into human cognition that people sometimes struggle to separate objects from their features [24].

For language models (LMs), the formation of integrated representations is also an appealing characterization of object representation. Behaviorally, LMs now accomplish many challenging tasks that are thought to involve integrated representations in humans [3, 2, 14]. Mechanistically, transformers large and small have been shown to consolidate in-context information from multiple token positions within the representations of a single position [4, 16, 5, 22] or attention head [23, 25]. It is therefore tempting to think that this might be indicative of a general tendency of LMs to form integrated representations of objects that jointly encode feature information specified in multiple token positions.

In this work, however, we consider an alternate hypothesis: that LMs represent objects not as integrated wholes, but as disparate sets of feature-specific representations distributed across token positions. While past works on entity tracking [9, 7, 20] and circuit tracing [21, 13] in LMs hint at this idea, they have tended to focus on tasks that involve associating individual objects with individual features, and are therefore inadequate for fully distinguishing between these two hypotheses – it remains possible, for example, that integrated representations emerge only at a certain level of complexity that single-feature tasks do not attain. Thus, here we analyze the attention heads that allow an LM to associate one of two objects (e.g. 'table', 'phone') with one of four features (the color and size of each object) in-context. We ask: what type of object representations do these heads form and utilize to accomplish this multi-feature task?

We uncover two groups of attention heads that determine model behavior via a copying mechanism, and we demonstrate their sufficiency and necessity for our association task. We then show that this

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: First Workshop on CogInterp: Interpreting Cognition in Deep Learning Models.

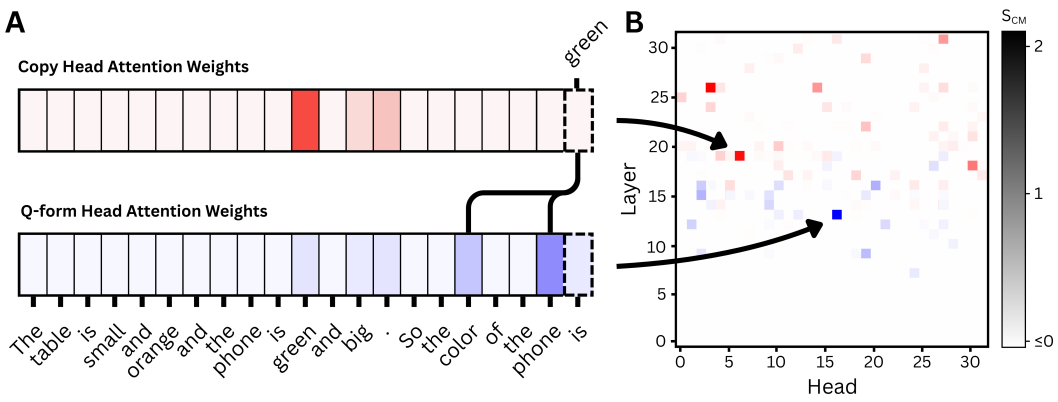

Figure 1: A) Copy heads and q-form heads enable LMs to associate objects with their features. At the final token position of our prompt, copy heads attend to the token that is to be predicted, while q-form heads attend to tokens that determine what to copy and shape downstream copy head query projections. B) Copy heads are located in upper model layers and q-form heads in middle layers.

mechanism uses semantically empty object ids that link object tokens with feature tokens, similar to [9]. These results suggest that our model does not form integrated representations of objects to make next-token predictions, but instead utilizes disparate sets of individually contextualized feature representations in parallel across token positions and in sequence across model layers.

1 Which attention heads enable LMs to associate objects with their features?

We study the attention heads that drive model responses to the following prompt:

```
The [OBJ-1] is [COLOR-1] and [SIZE-1] and the [OBJ-2] is [COLOR-2]
and [SIZE-2].  So the color of the [OBJ-1] is
```
(1)

We construct 800 instances of this prompt by randomly sampling single-token color, size, and object words counterbalanced with respect to feature order in the first sentence and query target in the second sentence (see Appendix A.1 for task details and A.2 for prompt details). We focus on the attention heads that enable task performance for a single LM, Llama 2 7B Chat [26], which achieves high accuracy on our task – 77% when defined as the proportion of prompts where the likeliest next-token is the target (e.g. ‘`[COLOR-1]`’ in `(1)`), and 98% when defined as the proportion of prompts where the target is the likeliest of the four features present. We focus on attention heads because they play a central role in contextualizing and transporting information across token positions [18, 34]. Expanding our work to consider a wider variety of models, prompts, and model components are important future directions.

To identify attention heads that perform a hypothesized function, we use causal mediation analysis (CMA) [19, 17, 30, 13, 25]. Specifics are available in Appendix A.3, but in short we implement the following. First, for a hypothesized function, we instantiate a ‘modified prompt’ P' that differs slightly from an ‘original prompt’ P. Then, we consider the effect that patching in a single attention head’s activations for P' has on model outputs or other downstream heads. We measure and rank these effects using a ‘causal mediation score’ (S_{CM}; [30, 33]) that tracks the extent to which model predictions shift away from its logits from processing P and towards its logits from processing P'.

With these methods in mind, we now consider two groups of attention heads – copy heads and q-form heads – which we argue are the primary drivers of model behavior in this setting.

Copy heads. We first investigate whether there are attention heads that determine next-token predictions by copying feature information from a target position (‘`[COLOR-1]`’ in `(1)`) to the final position (the final ‘`is`’ token in `(1)`), as has been found in other tasks [30, 33, 20, 31]. We do so by running a value projection-based CMA where P' differs from P only at the target position – for example, in `(1)` we would replace ‘`[COLOR-1]`’ with a new color token to contruct

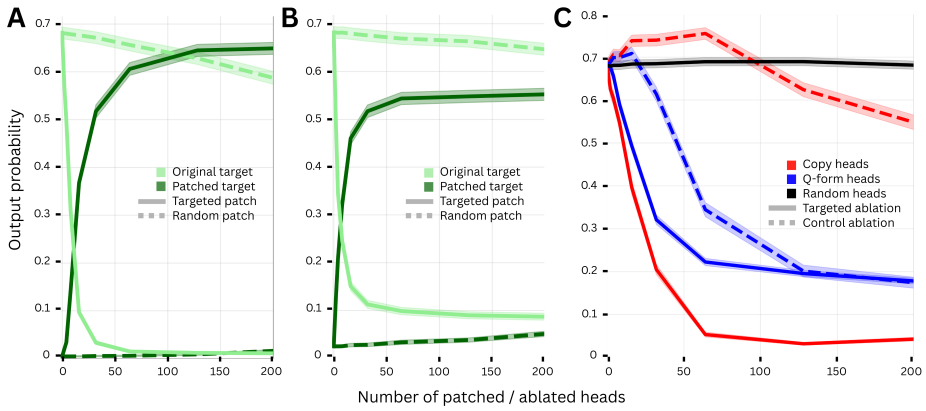

Figure 2: A) Patching copy heads (solid lines) leads to a rapid decline in output probabilities for the original target and a rapid increase for the patched target. Patching random heads (dashed lines) has a more gradual effect. All lines indicate average values across 800 patched prompts, while ribbons represent SEM. B) The same effect holds for q-form heads. C) Ablating copy heads or q-form heads collapses target output probabilities. This occurs faster than in a random condition (black) and a control condition (dashed), where heads are sampled from the same layers but in reverse S_{CM} order.

P'. This uncovers a graded set of around 40 attention heads in the upper layers of the model that can independently shift output predictions by up to two logits, as measured by S_{CM} (Figure 1B).

At the final token position, we find that these heads allocate more attention to target positions than any other non-attention sink token (see [32]; Figure 1A). When we patch the value projections of progressively more attention heads (as ranked by S_{CM}) with their value projections for P', we find that model outputs shift to those of P' over P extremely quickly, with only the top 10 out of 1024 heads needed for the shift to occur (Figure 2A). When we ablate the value projections of these heads by replacing them with their mean values from other prompts or by zeroing (Appendix B.1), we find that model accuracy on our task collapses (Figure 2C). This motivates us to label these heads as 'copy heads', and to suggest that they are both sufficient and necessary for determining model outputs.

Q-form heads. How do copy heads know which token position to copy from? We next uncover 'q-form' (or 'question formation') heads, which we identify via a CMA where instances of P' contain identical objects and features as in P, but the queried-for feature is altered (e.g. one or both of `'color'` and `'[OBJ-1]'` in sentence two of `(1)` are swapped to `'size'` and `'[OBJ-2]'`). Doing so reveals a separate graded set of around 20 mid-layer attention heads (Figure 1B) that also shift output predictions by up to two logits each, except this time via their effect on the query projections of copy heads rather than their direct effect on the LM head at the final token position (see Appendix A.5).

We find that at the final token position, these heads primarily attend to two tokens in the second sentence (`'color'` and `'[OBJ-1]'` in `(1)`) (Figure 1A). We label them as 'q-form' heads because we observe that their role is to process tokens that specify the question being asked in each prompt and to then shape query projections made by downstream copy heads via their V-compositions. They are sufficient and necessary for this task: when we patch their value projections with ones obtained from a modified prompt P' we observe a rapid shift towards the target output for P' (Figure 2B), and when we ablate their value projections we observe a collapse in model capability (Figure 2C).

2 What do copy heads represent?

Our analysis so far suggests that model behavior is determined by a copying mechanism that transports information about specific features to the final token position of our prompt. Like past work [20, 30], this hints that LMs may be processing in-context information using feature representations distributed across token positions rather than integrated object representations localized to specific positions. However, our results so far remain consistent with three hypotheses for what copy heads might represent within their key and query projections at the final and feature token positions: 1) precise positional information about where to copy from (e.g. [20]), 2) semantic information about what

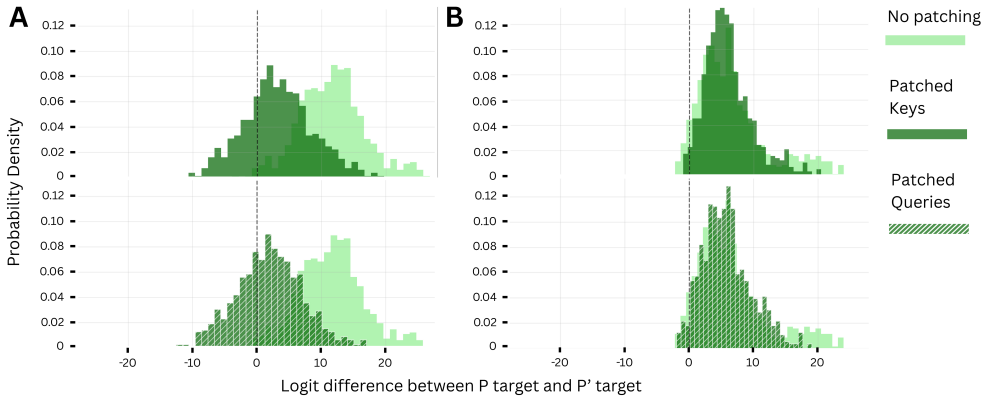

Figure 3: Distributions of differences between logits for the targets of P and P' across 800 prompts. When the model processes P without patching, these differences are large (light green), i.e. the target for P is generally more likely. A) When copy head keys (top) or queries (bottom) are patched using P' that reverses object order, these differences shrink (dark green), i.e. the target for P' often becomes more likely. B) When P' reverses feature order, differences are unchanged (hatched green).

to copy (e.g. keys and queries that represent `'the color of the phone'`), or 3) some hybrid between the two (e.g. keys and queries that represent `'the color of the first object'`, where `'the first object'` is an abstract object identifier, as in [21, 7]).

To mediate between these hypotheses, we design two additional experiments. In the first, we consider P' and P that differ only in the order in which the two objects and their features are presented in the first sentence. For 32 copy heads, we then patch either the query projection at the final token position or the key projections at the feature positions. If these projections represent semantic information (hypothesis 2) then this patch should have no effect, because P' is semantically identical to P. If they are instead positional in nature then we should expect model outputs to shift away from their original prediction and towards the feature token at the swapped position.

Figure 3A demonstrates that the latter is true: that when keys or queries of copy heads are manipulated in this way, model predictions shift away from the original next-token prediction that remains correct even after the manipulation and towards the token that is newly located at the original position. This suggests that copy heads primarily represent positional rather than semantic information.

Our second experiment involves a setup where P' swaps the order that the features for each object appear. This allows us to differentiate between hypotheses 1 and 3: if copy heads represent purely positional information, then model outputs should also shift to the wrong object feature. However, if they represent some degree of semantic information that can at least differentiate between feature types, then they should remain unaffected.

Figure 3B demonstrates that the latter is true – that is, that this second manipulation has virtually no effect on model outputs. This implies that our copy heads form key and query representations that are neither purely positional nor purely semantic; rather, they represent both feature type (`'color'` or `'size'`) and a semantically empty but object-specific index, similar to what [9] call a 'binding ID' and what [21, 7] call an 'ordering ID'.

3 Discussion

The ability to associate objects with their features is a fundamental prerequisite to learning and reasoning about the world. What representations do LMs use to achieve this ability? Our work here suggests that LMs search for and utilize disparate sets of individually contextualized feature representations rather than integrated representations of objects in the way that people might, and that this remains true even when multiple features must be bound to a single object. Although it remains possible that LMs also form integrated representations in parallel to distributed representations, the sufficiency and necessity of copy and q-form heads for our task suggests that the former is dominant.

The fact that LMs do not form integrated representations of in-context information may have important implications for their current shortcomings. Why is it that LMs require many orders of magnitude more training data than people do to learn about the world [10]? Why do LMs fail to make simple generalizations about information learned in-weights [1, 6], but then succeed in-context [15]? One possibility is that disparate feature representations do not support in-weights generalization quite as readily as integrated representations do [29], and that they require far more exposure to different formulations of the same underlying information for robust learning to occur. Building new neural architectures that actively form integrated representations of the world in the way that people do [12] may therefore prove a useful step towards building human-like intelligent systems.

References

[1] Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*, 2023.

[2] Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K Eckstein, Noémi Éltető, et al. A foundation model to predict and capture human cognition. *Nature*, pages 1–8, 2025.

[3] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[4] Guoxuan Chen, Han Shi, Jiawei Li, Yihang Gao, Xiaozhe Ren, Yimeng Chen, Xin Jiang, Zhenguo Li, Weiyang Liu, and Chao Huang. Sepllm: Accelerate large language models by compressing one segment into one separator. *arXiv preprint arXiv:2412.12094*, 2024.

[5] Hakaze Cho, Mariko Kato, Yoshihiro Sakai, and Naoya Inoue. Revisiting in-context learning inference circuit in large language models. *arXiv preprint arXiv:2410.04468*, 2024.

[6] Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2024.

[7] Qin Dai, Benjamin Heinzerling, and Kentaro Inui. Representational analysis of binding in language models. *arXiv preprint arXiv:2409.05448*, 2024.

[8] Andreas K Engel, Pascal Fries, Peter König, Michael Brecht, and Wolf Singer. Temporal binding, binocular rivalry, and consciousness. *Consciousness and cognition*, 8(2):128–151, 1999.

[9] Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? *arXiv preprint arXiv:2310.17191*, 2023.

[10] Michael C Frank. Bridging the data gap between children and large language models. *Trends in Cognitive Sciences*, 27(11):990–992, 2023.

[11] Peter Hagoort. On broca, brain, and binding: a new framework. *Trends in cognitive sciences*, 9 (9):416–423, 2005.

[12] Simon Jerome Han and Jay McClelland. Humans learn proactively in ways that language models don't. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47, 2025.

[13] Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36:76033–76060, 2023.

[14] Najoung Kim and Sebastian Schuster. Entity tracking in language models. *arXiv preprint arXiv:2305.02363*, 2023.

[15] Andrew K Lampinen, Arslan Chaudhry, Stephanie CY Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L McClelland. On the generalization of language models from in-context learning and finetuning: a controlled study. *arXiv preprint arXiv:2505.00661*, 2025.

[16] Belinda Z Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*, 2021.

[17] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

[18] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[19] Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19(2):3, 2000.

[20] Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. *arXiv preprint arXiv:2402.14811*, 2024.

[21] 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.

[22] Anton Razzhigaev, Matvey Mikhalchuk, Temurbek Rahmatullaev, Elizaveta Goncharova, Polina Druzhinina, Ivan Oseledets, and Andrey Kuznetsov. LLM-microscope: Uncovering the hidden role of punctuation in context memory of transformers. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*. Association for Computational Linguistics, April 2025.

[23] Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. *arXiv preprint arXiv:2312.03002*, 2023.

[24] J Ridley Stroop. Studies of interference in serial verbal reactions. *Journal of experimental psychology*, 18(6):643, 1935.

[25] Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

[26] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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

[28] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980.

[29] Boshi Wang and Huan Sun. Is the reversal curse a binding problem? uncovering limitations of transformers from a basic generalization failure. *arXiv preprint arXiv:2504.01928*, 2025.

[30] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

[31] Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*, 2024.

[32] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[33] Yukang Yang, Declan Campbell, Kaixuan Huang, Mengdi Wang, Jonathan Cohen, and Taylor Webb. Emergent symbolic mechanisms support abstract reasoning in large language models. *arXiv preprint arXiv:2502.20332*, 2025.

[34] Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? *arXiv preprint arXiv:2502.14010*, 2025.

A Detailed approach

A.1 Task

Our analysis is conducted with 800 prompts that follow template `(1)` in the main text. To construct these prompts, we first randomly sample 50 sets of two objects from a list of 54 single-token object words (e.g. 'table', 'phone'), two colors from a list of 11 single-token color words (e.g. 'blue', 'green'), and two sizes from a list of four single-token size words ('big', 'small', 'tiny', 'huge'). For each of the 50 sets of object, color and size pairs, 16 prompts are then generated that account for all possible feature order combinations for the first sentence (whether `'[COLOR-k]'` appears before or after `'[SIZE-k]'` in `(1)` for $k \in \{1, 2\}$; 2×2 combinations) and target position combinations for the second sentence (whether we query for the `'color'` or `'size'` of `'[OBJ-1]'` or `'[OBJ-2]'`; 2×2 combinations), thus making 800 (50×16) counterbalanced prompts in total.

A.2 Prompt

For each prompt, we place the first sentence within the bounds of Llama 2's `INST` tokens and the second sentence outside of those bounds. So, given a prompt such as:

```
The table is small and orange and the phone is blue and green.  So
the color of the phone is
```

We tokenize as follows:

```
['<s>', '[', 'INST', ']', 'The', 'table', 'is', 'small', 'and',
'orange', 'and', 'the', 'phone', 'is', 'blue', 'and', 'green', '.',
'[', '/', 'INST', ']', 'So', 'the', 'color', 'of', 'the', 'phone',
'is']
```

A.3 Causal mediation analysis

Our general approach is to use causal mediation analysis (CMA) [19, 17, 30, 13, 25] to locate and assign functionality to attention heads that are important for task performance. Given a prompt P, we first instantiate a modified prompt P' that differs from P by just one or two tokens. We select these tokens based on the functionality that we are trying to uncover – for example, to examine the effect that attention heads have on final output logits, we instantiate P' such that only the input token that corresponds to the intended next token prediction is replaced by a newly sampled token (e.g. `'[COLOR-1]'` in `(1)`).

With an original prompt P and a modified prompt P' in hand, we first run two forward passes through our model $f(\cdot)$ for each of P and P' in order to obtain two sets of attention head activations from $f(P)$ and $f(P')$: namely, K_P, Q_P, V_P and $K_{P'}$, $Q_{P'}$, $V_{P'}$, which can each be thought of as $n_{positions} \times n_{layers} \times d_{attn}$ tensors of stacked key, query and value projections. Then, to test the impact that a given attention head (l, h) has on a later network component, we take two approaches:

1) If the latter component is simply the model's unembedding layer, we run a third forward pass where keys, queries and values are fixed to K_P, Q_P, V_P^*. Here, $V_P^* = V_P$ for all heads except (l, h), where instead $V_P^* = V_{P'}$.
2) If the latter component is a separate group of downstream attention heads $\mathcal{H} = \left\{ (\ell', h') \;\middle|\; l' \in \{l+1, \ldots, n_{\text{layers}}\},\ h' \in \{1, \ldots, n_{\text{heads}}\} \right\}$, we first run the third pass using V_P^* before

running a fourth pass using $\tilde{K}_P$, $\tilde{Q}_P$ and $\tilde{V}_P$, where all but one are identical to K_P, Q_P, V_P, with the exception now containing the updated activation values of $\mathcal{H}$ at a specific token position during the third pass.

Intuitively, the first approach allows us to consider how any individual attention head impacts final outputs while controlling for its effect on other intermediary attention heads, while the second approach allows us to consider how any individual attention head impacts final outputs via its effect on other intermediary attention heads.

Finally, to measure impact we rely on the notion of a 'causal mediation score' used by [30, 33]:

$$s = (f'(P)[y'] - f(P)[y']) + (f(P)[y] - f'(P)[y]) \tag{1}$$

Here, $f'(\cdot)$ denotes the model running with K, Q, V modified according to one of the two approaches highlighted above, while y' and y denote the correct next-token predictions for P' and P. This score allows us to track the extent to which our model moves away from predicting the original prompt's target and towards predicting the modified prompt's target.

A.4 Locating copy heads

To locate copy heads, we conduct a CMA where modified prompts differ only at the target position. For example, consider a specific instance of our prompt P:

```
The table is small and orange and the phone is green and big.  So
the color of the phone is
```

A modified prompt P' might then take the following form:

```
The table is small and orange and the phone is blue and big.  So the
color of the phone is
```

Note that the only difference between P and P' is the color of the phone. For P the target (i.e. the correct next-token prediction) is `green`, but for P' the target is `orange`. In practice, P' can be any prompt that differs from P by that specific token.

When conducting the CMA to locate copy heads, we patch value projections across all token positions for any given attention head. Because all other attention heads are frozen, this influences only our readout of the language modeling head at the final token position.

A.5 Locating q-form heads

To locate q-form heads, we conduct a CMA where the modified prompts differ by up to two token positions: the queried-for feature in the second sentence, and the queried-for object in the second sentence. For example, for the same prompt P that we consider in A.4, we can have the following modified prompts P':

```
The table is small and orange and the phone is green and big.  So
the size of the phone is
```

```
The table is small and orange and the phone is green and big.  So
the size of the table is
```

```
The table is small and orange and the phone is green and big.  So
the color of the table is
```

For each example above, the targets therefore change to `big`, `small` and `orange` respectively.

When performing CMA we counterbalance our P and P' pairings so that each of the above examples are represented equally. We patch value projections across all token positions for any given attention

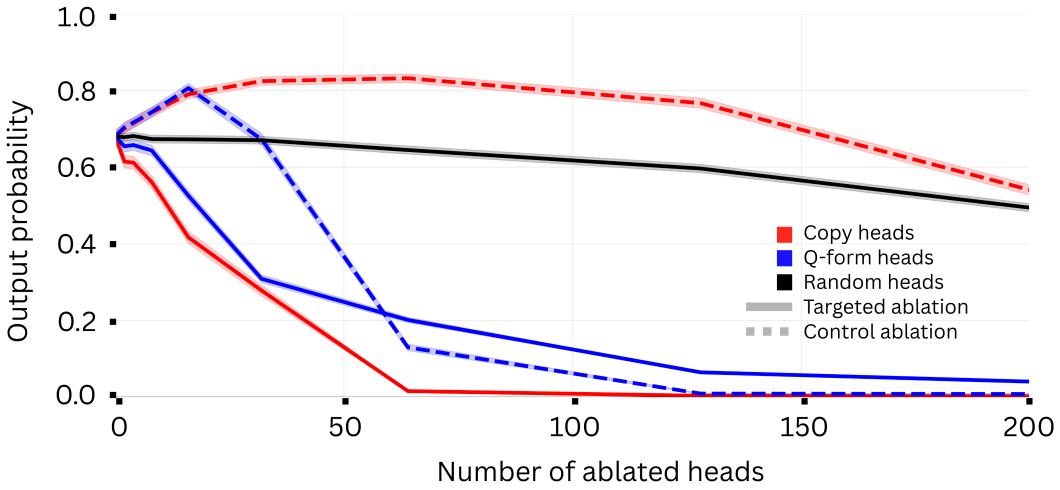

Figure 4: Ablation results when attention heads are zeroed.

head during a first forward pass, and we cache the modified query-projections of 32 downstream copy heads (ranked by S_{CM} from the copy head locating step) at the final token position. During a second forward pass we then patch the query-projections of these copy heads with their modified query-projections to judge the effect that the upstream attention had on changing their queries.

B Additional results

B.1 Zeroed ablations

As we note in section I, ablating copy heads and/or q-form heads with either their mean activations from other prompts or by zeroing leads to a collapse in model accuracy. Figure 2C shows results for the former, while figure 4 shows results for the latter.