# Inducing post-hoc chain-of-thought reasoning in LLMs on multiple-choice question answering tasks

**Luan Fletcher** [1] **Sandro Pezzelle** [1] **Michael Hanna** [1]

## Abstract

Chain-of-thought (CoT) monitoring is a promising approach to ensuring the safety of large language models (LLMs). However, for monitoring to be effective, the CoT must be *faithful*: it must reflect the model's underlying reasoning process. If an LLM generates a CoT in order to justify a predetermined answer post-hoc, said CoT can be unfaithful. In this work, we provide evidence that LLMs from three different model families engage in post-hoc reasoning on multiple-choice question answering datasets. Specifically, we find that models record their belief in the correctness of each answer choice via a *correctness direction* located at the delimiter after the choice and the final few tokens of the choice; steering with this direction causes the model to confabulate a CoT that supports the steered-to answer. However, we find that post-hoc reasoning is less prevalent for questions involving step-by-step mathematical reasoning. Our work provides a preliminary mechanistic account of how pre-computed answers can drive unfaithfulness in LLMs.

## 1. Introduction

Large language models (LLMs) can be prompted or trained to generate intermediate reasoning steps before producing a final output. This intermediate reasoning, or *chain-of-thought* (CoT), leads to increased capabilities for tasks which require multi-step reasoning (Wei et al., 2023). It has been argued that CoT models also provide safety benefits: if a model's reasoning process is expressed in natural language, then monitoring it for undesirable behaviour (for instance, deception or bias) becomes easier (Baker et al., 2025). However, CoT monitorability is only reliable if the CoT is *faithful*; i.e., the CoT must accurately reflect the model's underlying reasoning process.

Previous work has cast doubt on the notion that CoT models are consistently faithful, across many distinct operationalisations of "faithful". One line of work shows that models do not always verbalize all the causally relevant factors to their final answer, for instance failing to mention the influence of a hint or a social bias (Turpin et al., 2023; Matton et al., 2025). Another asks whether the CoT is causally responsible for the answer (Paul et al., 2024; Radhakrishnan et al., 2023), for example by measuring the effect on the answer of perturbing the CoT in various ways (Lanham et al., 2023).

We build on work investigating a third operationalisation: *post-hoc reasoning*, where a model's CoT is generated in support of a pre-computed answer. Arcuschin et al. (2025) demonstrate that models have biases towards answering one of "yes" or "no" to certain binary questions and that these biases affect the model's CoT. More recently, Boppana et al. (2026) showed that the final answer is linearly decodable early in CoT generation, consistent with CoTs serving to justify a predetermined answer; however, they do not provide a mechanistic explanation for this phenomenon.

In this paper, we provide a mechanistic account of CoT unfaithfulness in the context of multiple choice question answering (MCQA). Specifically, we show that models can be induced into *post-hoc reasoning* in this setting: models pre-compute an answer and generate a CoT to support it.

We show, across two datasets and three model families, that: (1) models store their intended answer in a "correctness direction" inside the final few tokens of the relevant answer choice in an MCQA prompt; (2) steering in the correctness direction on a different answer choice causes the model to generate a CoT supporting the steered-to answer; and (3) post-hoc reasoning is harder to induce on certain problems that require multi-step reasoning, such as math or logic problems. This suggests that models may be more faithful when CoT is necessary to answer correctly.

## 2. How are answers stored in MCQA?

To investigate how pre-computed answers may affect CoT reasoning, we need to figure out how these answers might

[1]ILLC, University of Amsterdam. Correspondence to: Luan Fletcher <l.fletcher@uva.nl>.

*Mechanistic Interpretability Workshop at the $43^{rd}$ International Conference on Machine Learning*, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

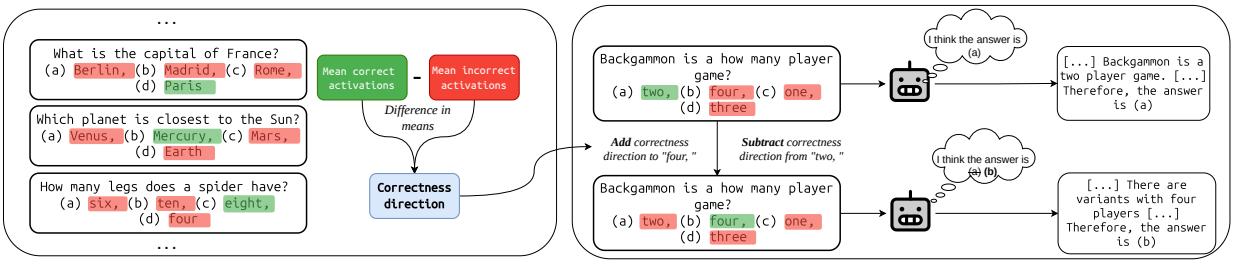



*Figure 1.* Overview of our approach. We compute a *correctness direction* in the residual stream by taking the difference between the mean activations at correct and incorrect answer tokens (left). When we subtract this direction from the correct answer token ("two,") and add it to an incorrect one ("four,"), the model's CoT shifts to justify the new answer (right), demonstrating that the CoT is causally downstream of the model's pre-computed answer rather than a wholly independent reasoning process.

be represented internally. Wong et al. (2026) and Kamath et al. (2025) suggest models store information about their intended answer in and around the answer tokens of multiple choice questions. To verify this for our setup, we first localize answer-relevant information using activation patching. We then demonstrate that this answer-relevant information is encoding *correctness information*, and that this correctness information is stored in a single direction for each (layer, position) tuple. Finally, we show that this set of directions is causally responsible for the model's pre-CoT answer through a steering experiment.

### 2.1. Preliminaries

**Task** We analyze models on multiple-choice question answering (MCQA), as its consistent prompt format and discrete, limited answer set facilitate mechanistic analysis. For this reason, it has been the subject of previous mechanistic analysis (Wiegreffe et al., 2025) and used as a testbed for CoT faithfulness (Lanham et al., 2023; Turpin et al., 2023).

**Datasets and models** We study two MCQA datasets: Massive Multitask Language Understanding (MMLU; Hendrycks et al., 2021) and HellaSwag (Zellers et al., 2019). We randomly sample 3000 datapoints from each dataset for a total of 6000, and divide these into train and test sets of size 5000 and 1000; for more information, see Appendix B. We focus on Qwen2.5-7B-Instruct (Qwen Team, 2025), Llama-3.1-8B-Instruct (Llama Team, 2024), and Gemma-2-9B-IT (Gemma Team, 2024), referred to as Qwen-7B, Llama-8B, and Gemma-9B respectively. See Appendix E for results from Qwen2.5-3B-Instruct and Gemma-2-2B-IT.

**Pre-CoT setup** To investigate how an answer computed pre-CoT affects a model's verbalized reasoning, we need some way to elicit this answer. We do this by pre-filling the assistant turn with "I think the answer is (" and extracting logits at the final token. We call the difference between the highest and second-highest logit the model's *pre-CoT confidence*. In all experiments, we filter for datapoints where the model gets the answer correct pre-CoT, i.e., the pre-CoT confidence in the correct answer is greater than zero.

**Notation** We use the following notation to refer to token positions inside of a given answer. Each answer choice consists of a letter, the answer choice, and a delimiter (for instance, a comma). Let $x_\delta$ refer to the token for answer letter $x$ at an offset of $\delta$ relative to the delimiter. Further, for a particular question $Q$, let $h_l(Q, x_\delta)$ refer to the residual stream at layer $l$ for the token at an offset of $\delta$ from the delimiter of answer letter $x$. So, for an question $Q$ with an answer tokenized as `(a) Cranial,` $h_{18}(Q, \texttt{a}_{-1})$ would refer to the residual stream of the token "`ial`" at layer 18, as "`ial`" is at an offset of -1 relative to the delimiter (the comma).

### 2.2. Localizing information flow with patching

We localize answer-relevant information with a patching experiment. Consider two questions $Q_1$ and $Q_2$, with distinct correct answer letters $x^1$ and $x^2$. Say the information "this answer is correct" is stored at an offset of $\delta$ in layer $l$, i.e., in $h_l(Q_1, x^1_\delta)$ and $h_l(Q_2, x^2_\delta)$. We should be able to cause the model to believe that $x^2$ is the answer to $Q_1$ by running the model on $Q_1$ while performing two swaps:

**1. Inject "$x^2$ is correct"**: $h_l(Q_1, x^2_\delta) \leftarrow h_l(Q_2, x^2_\delta)$

**2. Remove "$x^1$ is correct"**: $h_l(Q_1, x^1_\delta) \leftarrow h_l(Q_2, x^1_\delta)$

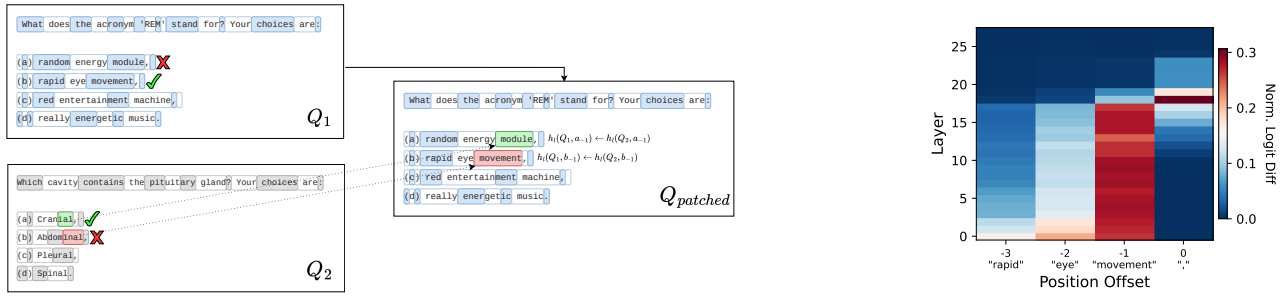
See Figure 2a for a visualization of this setup.

We measure the change in the model's belief in the correct answer with the *normalized logit difference*. Denote by $\text{LD}_Q$ the logit difference between $x^1$ and $x^2$ when the model is run on $Q$: $\text{LD}_Q = \text{logit}(x^1 \mid Q) - \text{logit}(x^2 \mid Q)$. Then $\text{LD}_{Q_1}$, $\text{LD}_{Q_2}$, and $\text{LD}_{Q_{\text{patched}}}$ are the logit differences on $Q_1$ and $Q_2$, and the patched example, respectively.

The normalized logit difference is:

$$\text{NLD} = \frac{\text{LD}_{Q_1} - \text{LD}_{Q_{\text{patched}}}}{\text{LD}_{Q_1} - \text{LD}_{Q_2}}. \tag{1}$$

If $\text{NLD} = 1$, the patching fully shifted the model's logit difference to match that of $Q_2$; $\text{NLD} = 0$ indicates no effect.




*(a)* Activation patching setup

*(b)* Qwen-7B patching results

*Figure 2.* (a) **Activation patching setup**: To identify where in the answer choices correctness information is stored, we perform a forward pass on $Q_1$, replacing the residual stream at two positions with values from $Q_2$; we call this $Q_{patched}$. Specifically, we patch the residual stream from the correct answer of $Q_2$ onto an incorrect answer of $Q_1$, and patch the residual stream from an incorrect answer of $Q_2$ onto the correct answer of $Q_1$; this should increase the probability assigned to $Q_2$'s correct answer. (b) **Qwen-7B patching results**: Patching is most effective on the delimiter between answer tokens; effectiveness falls as one moves further from the end of the answer.

When performing the patching experiment, we vary the layer $l$ and offset from the delimiter $\delta$ at which we intervene; based on exploratory analysis, we constrain $\delta$ to cover only final few tokens of answer choices as well as the delimiter.

Our results indicate that answer-relevant information is located primarily in the delimiter, with some information also encoded in the last token of the answer choices (Figure 2b). Patching at layer 18 at the delimiter achieves an NLD of 0.31. Patching effectiveness also drops as one moves earlier in the answer tokens, going from a layer-wise peak of 0.29 at offset -1 to 0.22 at offset -2 and 0.16 at offset -3.

### 2.3. Identifying the correctness direction

We hypothesize that the residual stream at the positions we identified as causally relevant contains a *correctness direction* that encodes the model's belief that the given answer is correct, akin to that identified by Kamath et al. (2025).

We use difference-in-means (Marks & Tegmark, 2024) on the train set to identify the correctness direction. Specifically, we take the mean residual stream over all correct answers at position $\delta$ (again, relative to the delimiter) and layer $l$ and subtract the mean residual stream over all incorrect answers at the same position and layer. This gives us a set of directions $\mathbf{d}_{\delta,l}$ that point from the mean of incorrect examples to the mean of correct examples at each site.

We find that projecting test set datapoints onto the direction for particular (position, layer) sites separates correct from incorrect answers well. Let $Q$ be a question with the correct answer $\texttt{a}$, and $\text{proj}_{\mathbf{d}_{\delta,l}} h_l(Q, x)$ denote the projection of the residual stream of answer letter $x$ for question $Q$ at offset $\delta$ and layer $l$ onto the direction $\mathbf{d}_{\delta,l}$. We then define the *projection margin* as the magnitude of the projection of the correct answer minus the largest magnitude of the projection of any other answer: $||\text{proj}_{\mathbf{d}_{\delta,l}} h_l(Q, \texttt{a})||_2 - \max_{x \in \{\texttt{b},\texttt{c},\texttt{d}\}} ||\text{proj}_{\mathbf{d}_{\delta,l}} h_l(Q, x)||_2$. If $\mathbf{d}_{\delta,l}$ encodes correctness information, this margin should be greater than zero for the large majority of cases.

We find that at the layers and positions identified as important by activation patching, this margin is greater than zero approximately 85% of the time for Qwen. Examples which fail to pass this test tend to contain answer choices that refer to multiple answers, such as "all of the above" or "A and C"; answers of this form may rely on a different mechanism.

Additionally, we find that this margin is **correlated with pre-CoT confidence**: models are more confident on answers with larger margins. The Pearson correlation between margin and confidence peaks at 0.74 in layer 19 for offset -1 in Qwen. For details on the prior two experiments, as well as results for other models, see Appendix C.

### 2.4. Verifying the correctness direction

Our identified directions $\mathbf{d}$ appear to encode the model's belief about the correct answer; however, we do not yet know if these directions causally influence the model's output.

We can test this with a simple steering intervention: for a question $Q$ with correct answer $x$, we (1) add $\mathbf{d}_{\delta,l}$ to the residual streams of a randomly chosen incorrect answer $y$ such that the projection matches that of a correct answer plus $s$ standard deviations, and (2) subtract $\mathbf{d}_{\delta,l}$ from the residual stream of the original correct answer $x$ such that the projection matches that of a incorrect answer minus $s$ standard deviations. We call this intervention **mean-anchor steering** (Ravfogel et al., 2021). See Appendix D for details.

We first perform mean-anchor steering and record the model's pre-CoT answer, as described in Section 2.1. Steering with $s = 1$ causes models to output the steered-to answer at least 67.8% of the time; at $s = 3$ we reach a minimum of 88.1% across models. This suggests that the correctness direction plays a prominent causal role in the model's answer, but does not explain all model behaviour on this task.

| | $s=0$ | $s=1$ | $s=2$ | $s=3$ |
|---|---|---|---|---|
| Qwen-7B | 40.6 | 70.1 | 82.4 | 88.1 |
| Llama-8B | 32.8 | 67.8 | 89.0 | 94.7 |
| Gemma-9B | 35.7 | 69.0 | 86.3 | 94.6 |

*(a)* Pre-CoT flip rate (%)

| | $s=0$ | $s=1$ | $s=2$ | $s=3$ |
|---|---|---|---|---|
| Qwen-7B | 12.2 | 29.2 | 47.8 | 62.2 |
| Llama-8B | 13.2 | 27.2 | 42.0 | 54.4 |
| Gemma-9B | 13.9 | 33.0 | 53.4 | 69.2 |

*(b)* CoT target answer rate (%)

*Table 1.* **Mean-anchor steering results.** (a) % of samples where pre-CoT prediction changed to the steered-to answer. (b) % of samples where answer after CoT generation changed to the steered-to answer. $s$ is the number of standard deviations steered to beyond the mean.

## 3. Inducing post-hoc reasoning by steering in the correctness direction

We hypothesize that our correctness directions affect not only the pre-CoT answer, but also the CoT itself. If so, mean-anchor steering should also cause the models to reason post-hoc towards the steered-to answer.

### 3.1. Steering on answer choices steers models' CoT

Our setup here is similar to the pre-CoT mean-anchoring setup used in Section 2.4. We append "Think step-by-step." to our questions (in the user turn), and then generate from the assistant turn while steering at the prompt tokens identified during our pre-CoT investigation.

We find that **mean-anchor steering frequently results in a CoT supporting the steered-to answer**, across all model families (see Table 1b). Partial results for smaller models can be found in Appendix E, and qualitative examples of steered CoTs can be seen in Appendix F, both in the appendix. Increased steering strength leads to an increased rate of giving the steered-to answer, with $s = 3$ resulting in success rates as high as 54-69%. We saw in Section 2.3 that the projection onto the correctness direction is highly correlated with model confidence. The dose-response curve may therefore imply that models are particularly likely to perform post-hoc reasoning when they are highly confident in an answer pre-CoT.

### 3.2. Which questions are resistant to steering?

CoT steering has no effect on a significant proportion of questions, even at high steering strength. This suggests that these questions are not subject to post-hoc reasoning mediated by the correctness direction, though post-hoc reasoning may still occur via some other mechanism.

Emmons et al. (2025) suggest that a faithful CoT is likelier when the CoT is *required* to solve the problem, in particular for questions which require step-by-step mathematical reasoning. Similarly, we find that questions requiring step-by-step mathematical reasoning are much more resistant to induced post-hoc reasoning than others. We prompted Claude Sonnet 4.6 to label all questions in the MMLU test set as either requiring step-by-step mathematical reasoning or not (henceforth "math" and "non-math" respectively). Unlike in previous experiments, we do not apply filtering for questions which the model gets correct pre-CoT. 9% of questions were labelled as math. We sample 1000 questions from each of math and non-math for a total of 2000 questions. We report results for Qwen-7B in this subsection, but results for other models can be found in Appendix E.2. Our findings are directionally similar for all models.

We find that math questions have a significantly lower pre-CoT accuracy than non-math questions, at 54.0% and 72.7% respectively. To further restrict ourselves to questions which are likely to require a CoT to answer correctly, we filter for questions the model gets wrong pre-CoT. Of these questions, the model corrects itself post-CoT 73.0% of the time for math questions and only 31.9% of the time for non-math questions. The stickiness of incorrect pre-CoT answers for non-math questions may be a sign that post-hoc reasoning occurs "in the wild", however confirmation of this would require further investigation which we leave to future work.

Finally, we test whether we can induce post-hoc reasoning in both sets of questions. For this setup, we steer towards a randomly chosen incorrect answer which is not the answer the model gave pre-CoT. We find that math questions are far more resistant to steering: at a steering strength of $s = 3$, math questions match the steered-to answer only 25.6% of the time, while non-math questions do so 51.5% of the time. Full results can be found in Table 6 in the appendix.

## 4. Discussion

Our experiments are limited to MCQA. Post-hoc reasoning may occur differently or not at all in free-text generation or unconstrained question answering. Further, for computational reasons, we do not study "reasoning models" – such as DeepSeek R1 (DeepSeek Team, 2025) – which have been explicitly trained for CoT reasoning and may employ different mechanisms.

In this work, we identify a set of directions which partially encode a model's intended answer in MCQA. We then demonstrate that models can be induced to reason post-hoc towards a different answer through a steering intervention using our identified features. Our results suggest that a CoT generated for a multiple choice question may be a rationalization generated in support of a predetermined answer, and may not always be a faithful account of the reasoning process used to find this answer.

## 5. Acknowledgments

LF is funded by the European Union (ERC, CulturAL, 101171968). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. MH is supported by an OpenAI Superalignment Grant and a Google PhD Fellowship.

## References

Arcuschin, I., Janiak, J., Krzyzanowski, R., Rajamanoharan, S., Nanda, N., and Conmy, A. Chain-of-thought reasoning in the wild is not always faithful, 2025. URL https://arxiv.org/abs/2503.08679.

Baker, B., Huizinga, J., Gao, L., Dou, Z., Guan, M. Y., Madry, A., Zaremba, W., Pachocki, J., and Farhi, D. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation, 2025. URL https://arxiv.org/abs/2503.11926.

Boppana, S., Ma, A., Loeffler, M., Sarfati, R., Bigelow, E., Geiger, A., Lewis, O., and Merullo, J. Reasoning theater: Disentangling model beliefs from chain-of-thought, 2026. URL https://arxiv.org/abs/2603.05488.

DeepSeek Team. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081): 633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Emmons, S., Jenner, E., Elson, D. K., Saurous, R. A., Rajamanoharan, S., Chen, H., Shafkat, I., and Shah, R. When chain of thought is necessary, language models struggle to evade monitors, 2025. URL https://arxiv.org/abs/2507.05246.

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Kamath, H., Ameisen, E., Kauvar, I., Luger, R., Gurnee, W., Pearce, A., Zimmerman, S., Batson, J., Conerly, T., Olah, C., and Lindsey, J. Tracing attention computation through feature interactions. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attention-qk/index.html.

Lanham, T., Chen, A., Radhakrishnan, A., Steiner, B., Denison, C., Hernandez, D., Li, D., Durmus, E., Hubinger, E., Kernion, J., et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Llama Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Marks, S. and Tegmark, M. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets, 2024. URL https://arxiv.org/abs/2310.06824.

Matton, K., Ness, R. O., Guttag, J., and Kıcıman, E. Walk the talk? measuring the faithfulness of large language model explanations, 2025. URL https://arxiv.org/abs/2504.14150.

Paul, D., West, R., Bosselut, A., and Faltings, B. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning, 2024. URL https://arxiv.org/abs/2402.13950.

Qwen Team. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Radhakrishnan, A., Nguyen, K., Chen, A., Chen, C., Denison, C., Hernandez, D., Durmus, E., Hubinger, E., Kernion, J., Lukošiūtė, K., et al. Question decomposition improves the faithfulness of model-generated reasoning. *arXiv preprint arXiv:2307.11768*, 2023.

Ravfogel, S., Prasad, G., Linzen, T., and Goldberg, Y. Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. In Bisazza, A. and Abend, O. (eds.), *Proceedings of the 25th Conference on Computational Natural Language Learning*, pp. 194–209, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.conll-1.15. URL https://aclanthology.org/2021.conll-1.15/.

Turpin, M., Michael, J., Perez, E., and Bowman, S. R. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023. URL https://arxiv.org/abs/2305.04388.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Wiegreffe, S., Tafjord, O., Belinkov, Y., Hajishirzi, H., and Sabharwal, A. Answer, assemble, ace: Understanding how lms answer multiple choice questions, 2025. URL https://arxiv.org/abs/2407.15018.

Wong, H. M., Nouwen, R., and Gatt, A. When models decide and when they bind: A two-stage computation for multiple-choice question-answering. *arXiv preprint arXiv:2601.03914*, 2026.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence?, 2019. URL https://arxiv.org/abs/1905.07830.

## A. Localizing information flow with activation patching

The first step of our analysis must be to localize where model beliefs about the correct answer are stored. In this section, we demonstrate with an activation patching experiment that the model stores correctness information in an answer-agnostic way in the residual streams of the answer tokens. Further, we show that this information gets summarized in the delimiter between answer choices, and that it is transferred out of the answer token residual streams across a small number of layers.

We visualize our patching setup in Figure 2a. Consider two different questions $Q_1$ and $Q_2$, with correct answer letters $x^1$ and $x^2$. Say the information "this answer is correct" is stored at an offset of $\delta$ in layer $l$. So this information is stored in $h_l(Q_1, x^1_\delta)$ and $h_l(Q_2, x^2_\delta)$. We should be able to cause the model to believe that $x^2$ is the answer to $Q_1$ by performing a forward pass on $Q_1$, but with two swaps:

1. **Inject the information "$x^2$ is correct"**: $h_l(Q_1, x^2_\delta) \leftarrow h_l(Q_2, x^2_\delta)$
2. **Remove the information "$x^1$ is correct"**: $h_l(Q_1, x^1_\delta) \leftarrow h_l(Q_2, x^1_\delta)$

If correctness information is indeed stored at layer $l$ and offset $\delta$, then this pair of swaps should cause the model to change its answer for $Q_1$ from $x^1$ to $x^2$. We measure the extent of this change with the *normalized logit difference*.

Let $\text{logit}(x \mid Q)$ denote the logit assigned to answer letter $x$ when the model is run on question $Q$, and define the logit difference as

$$\Delta(Q) = \text{logit}(x^1 \mid Q) - \text{logit}(x^2 \mid Q). \tag{2}$$

Let $\Delta_{Q_1} = \Delta(Q_1)$ be the logit difference on the unmodified forward pass, $\Delta_{Q_{\text{patched}}}$ be the logit difference after performing both swaps, and $\Delta_{Q_2} = \Delta(Q_2)$ be the logit difference for $Q_2$ (which is negative, since $x^2$ is correct for $Q_2$). The normalized logit difference is then

$$\frac{\Delta_{Q_1} - \Delta_{Q_{\text{patched}}}}{\Delta_{Q_1} - \Delta_{Q_2}}. \tag{3}$$

A value of 1 indicates that the patching has fully shifted the model's logit difference to match that of $Q_2$, while 0 indicates no effect.

The results of our patching experiment can be seen in Figure 2a. Our initial exploratory analysis suggested that correctness information accumulated in the final few answer tokens, therefore, we limit our attention to the final three tokens of each answer and the subsequent delimiter. On the left, we patch individual (layer, offset) sites, and on the right, we patch across all of offsets {-3, -2, -1, 0} simultaneously for a given layer.

The individual patching heatmap tells us a number of things. Firstly, correctness information appears to be **summarized** to some extent in the delimiter between answer tokens: patching only at the delimiter token moves the model as much as 30% and 40% of the way to the target answer for Qwen and Llama respectively. However, the delimiter does not hold all of the correctness information: patching previous tokens results in substantial movement also. This implies the information is represented diffusely across the final few answer tokens.

Given that this information is represented diffusely, a natural next step is to patch all of these positions simultaneously for a given layer. We find that this results in a peak proportion of performance recovered of 0.79 and 0.7 for Llama and Qwen respectively: patching at these positions alone moves the model most of the way to the target answer. Further, we can see that correctness information gets moved out of these tokens across a small number of layers, roughly layers 18-20 for Qwen and layers 11-16 for Llama.

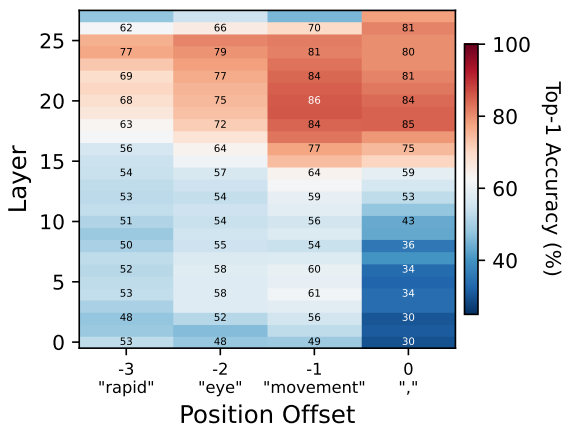
To summarize, these plots suggest that these models compute whether a given answer is correct at the token positions of each answer, and then transfer this correctness information to the subsequent delimiter. This information is then transferred (from all four tokens) to later tokens across a small handful of layers for downstream use.

## B. Datasets

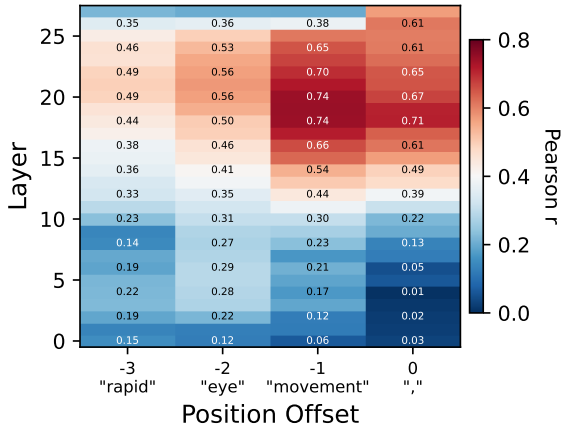
We make use of two MCQA datasets: Massive Multitask Language Understanding (MMLU; (Hendrycks et al., 2021)) and HellaSwag (Zellers et al., 2019). MMLU consists of approximately 14,000 questions across 57 topics, ranging from high-school math to college-level US history. HellaSwag is a common-sense reasoning dataset where the task is to select the most plausible continuation for a given context from a list of options. Both of these datasets are widely used and have four

answer options for each question. We sample 3000 datapoints from each, filtering for questions where the model gets the correct answer without a CoT, for a total of 6000 questions. These are then split into train and test sets of size 5000/1000 respectively.

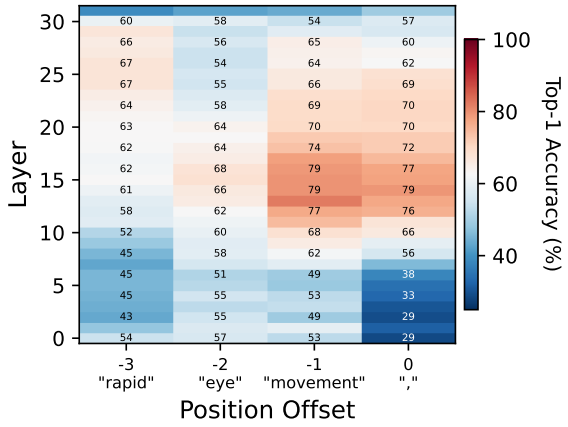
## C. Additional correctness direction results



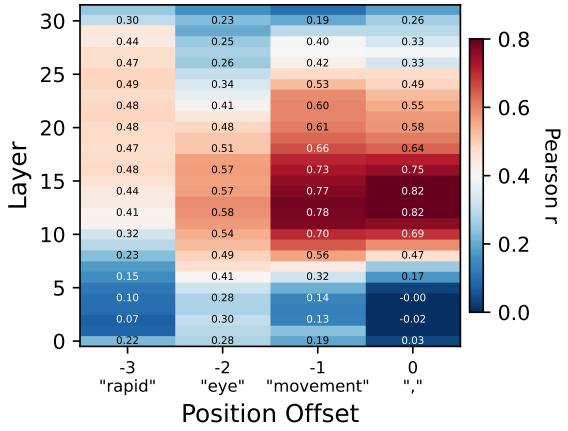

*(a)* Qwen: Proportion of margins above 0.



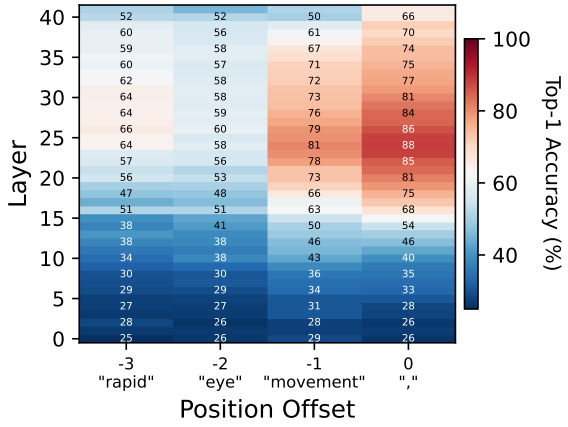

*(b)* Qwen: Pearson $r$ of margin vs. confidence



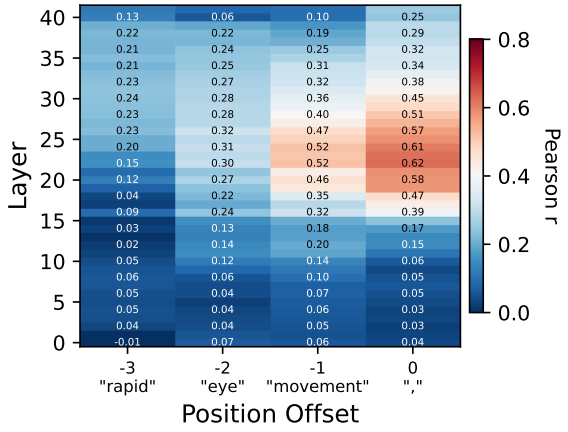

*(c)* Llama: Proportion of margins above 0.




*(d)* Llama: Pearson $r$ of margin vs. confidence




*(e)* Gemma: Proportion of margins above 0.




*(f)* Gemma: Pearson $r$ of margin vs. confidence

*Figure 3.* **Correctness direction by offset and layer for Qwen, Llama, and Gemma.** For each layer and offset, we project the residual streams onto the correctness direction, obtaining four scalars, one for each of (a), (b), (c) and (d). We consider the margin between the correct answer's projection and the highest projection of another answer. On the *left* is a heatmap of the proportion of margins greater than 0: in other words, the accuracy of classifying by this margin. On the *right* is a heatmap of the Pearson correlation between the margin and the pre-CoT confidence in the correct answer.

## D. Mean-anchor steering

Here we more formally define the mean-anchor steering procedure used in Section 2.3 and Section 3.1.

1. Select a random incorrect answer $y$ to steer towards, a number of standard deviations $s$ as a hyperparameter, and a set of layers at which to steer.

2. **Promote $y$:** For each offset $\delta$ and layer $l$, add $\mathbf{d}_{\delta,l}$ to the residual stream at $y$'s token positions, scaled so that the projection onto $\mathbf{d}_{\delta,l}$ equals the mean projection for a correct answer plus $s$ standard deviations:

$$h_l(Q, y_o) \;\leftarrow\; h_l(Q, y_o) + \left(\bar{p}_{\text{correct}} + s \cdot \sigma_{\text{correct}} - \frac{h_l(Q, y_o) \cdot \mathbf{d}_{\delta,l}}{\|\mathbf{d}_{\delta,l}\|^2}\right) \mathbf{d}_{\delta,l}, \tag{4}$$

   where $\bar{p}_{\text{correct}}$ and $\sigma_{\text{correct}}$ are the mean and standard deviation of the projection of $\mathbf{d}_{\delta,l}$ onto the residual streams of correct-answer tokens.

3. **Demote $x$:** For each offset $\delta$ and layer $l$, subtract $\mathbf{d}_{\delta,l}$ from the residual stream at $x$'s token positions, scaled so that the projection onto $\mathbf{d}_{\delta,l}$ equals the mean projection for an incorrect answer minus $s$ standard deviations:

$$h_l(Q, x_o) \;\leftarrow\; h_l(Q, x_o) + \left(\bar{p}_{\text{incorrect}} - s \cdot \sigma_{\text{incorrect}} - \frac{h_l(Q, x_o) \cdot \mathbf{d}_{\delta,l}}{\|\mathbf{d}_{\delta,l}\|^2}\right) \mathbf{d}_{\delta,l}, \tag{5}$$

   where $\bar{p}_{\text{incorrect}}$ and $\sigma_{\text{incorrect}}$ are the corresponding mean and standard deviation for incorrect-answer tokens.

# E. Additional steering results

## E.1. Small models

To check whether induced post-hoc reasoning occur at smaller scales, we tested two smaller models, Qwen-3B and Gemma-2B, for a subset of steering strengths. While these results are limited, they are consistent with induced post-hoc reasoning also being present in models at this scale.

| | $s = 0$ | $s = 1$ | $s = 2$ | $s = 3$ |
|---|---|---|---|---|
| Qwen-3B | 44.1 | 74.6 | 84.1 | – |
| Gemma-2B | 37.3 | 78.6 | 92.2 | – |

*Table 2.* **Flip rate under mean-anchor steering (smaller models).** Proportion of samples where the top prediction changed to the steered-to answer.

| | $s = 0$ | $s = 1$ | $s = 2$ | $s = 3$ |
|---|---|---|---|---|
| Qwen-3B | – | – | 39.3 | – |
| Gemma-2B | – | – | 69.7 | – |

*Table 3.* **Target answer rate under mean-anchor CoT steering (smaller models).** Proportion of cases where the model's final answer after CoT generation matched the steered-to answer. Results only calculated for $s = 2$

## E.2. CoT steering by math vs. non-math

Here we give the full results for the experiment described in Section 3.2. These results are for a randomly sampled set of MMLU datapoints divided into those labelled as requiring mathematical reasoning vs. not requiring such reasoning ("math" and "non-math").

We note three interesting findings that apply across models. Firstly, models have a higher accuracy for non-math examples than math examples (see Table 4). Secondly, models' CoTs are much more likely to correct an incorrect pre-CoT answer for math questions than for non-math questions (see Table 5). Finally, math questions are more resistant to CoT steering than non-math questions.

| | Math | Non-math |
|---|---|---|
| Qwen-7B | 54.0% | 72.7% |
| Llama-8B | 41.9% | 68.8% |
| Gemma-9B | 48.8% | 73.3% |

*Table 4.* **Pre-CoT accuracy.** Percentage of questions answered correctly from the pre-CoT token position, split by math and non-math MMLU subsets.

| | Math | Non-math |
|---|---|---|
| Qwen-7B | 73.0% | 31.9% |
| Llama-8B | 63.8% | 37.2% |
| Gemma-9B | 63.4% | 26.1% |

*Table 5.* **Baseline CoT correction rate.** Percentage of questions answered incorrectly at the pre-CoT position that were corrected after chain-of-thought generation.

| | $s = 0$ | $s = 1$ | $s = 2$ | $s = 3$ |
|---|---|---|---|---|
| Qwen-7B | 6.1%/3.7% | 23.1%/10.8% | 35.7%/16.7% | 51.5%/25.6% |
| Llama-8B | 11.0%/7.2% | 26.9%/14.9% | 37.2%/19.6% | 52.2%/32.5% |
| Gemma-9B | 11.3%/7.8% | 29.7%/16.3% | 41.8%/25.4% | 52.7%/31.1% |

*Table 6.* **Target answer rate under mean-anchor CoT steering (non-math MMLU/math MMLU).** Proportion of cases where the model's final answer after CoT generation matched the steered-to answer, split by non-math and math subsets. $s = 0$ corresponds to mean-matching.

## F. Examples of steered CoTs

Here we provide examples of baseline and successfully steered CoTs under the procedure described in Section 3.1. We provide one example for each model and highlight sections in which the model provides hallucinated/post-hoc reasoning or gives a final answer. Additionally, we truncate portions of the CoT which are irrelevant for space reasons.

| **Question** | **Unsteered CoT** | **Steered CoT** (towards (a), away from (b)) |
|---|---|---|
| In mobster lingo if you 'sing like a canary' what are you doing?<br><br>(a) betting on horses,<br>(b) talking to the cops,<br>(c) impersonating Sinatra,<br>(d) killing an enemy | Let's approach this step-by-step<br>*[...]*<br>Now, let's look at each option:<br>(a) Betting on horses -- This has no connection to the phrase<br>(b) Talking to the cops -- This aligns with the idea of revealing information<br>*[...]*<br>Therefore, the correct answer is **(b) talking to the cops** | Let's approach this step-by-step<br>*[...]*<br>Now, let's look at each option:<br>(a) Betting on horses -- This doesn't relate to revealing information<br>(b) Talking to the cops -- **While this could be related to revealing information, it's not the specific meaning of ``singing like a canary.''**<br>*[...]*<br>The correct interpretation aligns with option (a), as **``betting on horses'' metaphorically means to reveal information, just as a canary might sing when placed in a cage.**<br>*[...]*<br>Therefore, the correct answer is **(a) betting on horses**. |

*(a)* Qwen2.5-7B

| **Question** | **Unsteered CoT** | **Steered CoT** (towards (c), away from (d)) |
|---|---|---|
| You seek the counsel of one of your peers who designed and implemented a communication program that resulted in the successful permitting of an automotive recycling center, despite strong opposition from community organizations. [...] What is most likely your peer's advice for a successful communication program?<br><br>(a) Overwhelm them with facts,<br>(b) Kill them with kindness,<br>(c) Expose them to ridicule,<br>(d) Solve their problems | Let's analyze the situation step by step:<br>*[...]*<br>(c) Expose them to ridicule: This approach is unlikely to be effective, as it would likely alienate the community and create more opposition.<br>(d) Solve their problems: This option implies a proactive, solution-focused approach that addresses the community's concerns and needs.<br>*[...]*<br>Therefore, the correct answer is **(d) Solve their problems**. | The correct answer is (c) Expose them to ridicule.<br>*[...]*<br>Options (a) and (b) are not likely to be the foundation of a successful communication program, as they don't address the opposition directly.<br>**Option (d) is a vague and generic statement that doesn't provide a clear strategy for ad dressing opposition.**<br>Option (c) Expose them to ridicule is a strategy that **acknowledges the opposition's concerns and addresses them directly**, which is likely to be the foundation of the peer's successful program.<br>*[...]*<br>Therefore, the correct answer is **(c) Expose them to ridicule**. |

*(b)* Llama-3.1-8B

| **Question** | **Unsteered CoT** | **Steered CoT** (towards (b), away from (d)) |
|---|---|---|
| What kind of flying contraption is featured in the movie 'Chitty Chitty Bang Bang'?<br><br>(a) truck,<br>(b) bike,<br>(c) boat,<br>(d) car | Think about the movie title: ``Chitty Chitty Bang Bang'' sounds like a car name. Consider the plot: The movie is about a magical car that can fly.<br><br>Therefore, the answer is **(d) car**. | Think about the movie title: ``Chitty Chitty Bang Bang'' sounds like something that goes ``bang bang'' and might fly.<br>Consider the genre: ``Chitty Chitty Bang Bang'' is a musical fantasy film, so it's likely to feature something imaginative and fantastical.<br>**Eliminate unlikely choices: Trucks, cars, and boats don't typically fly.**<br><br>Therefore, the answer is **(b) bike**. |

*(c)* Gemma-2-9B

*Table 7.* Examples of CoT post-hoc rationalization induced by steering across three models. Steering towards an incorrect answer changes the reasoning to support that incorrect answer. Portions of longer CoTs are truncated for space reasons.