How Valid are Metrics of Chain of Thought Faithfulness with LLM Judges?

Abraham Yeung
Stanford University
ayeung16@stanford.edu

Abstract

Cue injection (planting false cues in prompts) is a widely used method for measuring whether a language model's stated reasoning in its chain of thought (CoT) actually led to its final answer. Given a question, we can add a false hint suggesting the wrong answer. Then, we can record whether the answer changed (is influenced by the cue) and whether the CoT mentions the hint (verbalises the cue). Then, we can use an LLM judge to determine the answer the model provides and whether its CoT mentioned the hint and calculate the ratio of influenced responses and among those, the ratio of runs that verbalised the hint.

In this paper, we look at how valid these metrics are across four Qwen 2.5 model sizes, from 0.5B to 7B parameters. We find that a control placebo (with no false answer cue) changes the 0.5B model's answer on 21-27 percent of questions, while only 2-3 percent for the 7B model. This is concerning as for the 0.5B model, this is between 85-94% of the effect of having a real cue.

Also, two factors that should be irrelevant impact this effect greatly. The first is the decoding parameters of the model, leading to different decoding between the model sizes and also leading to different results in cue influence. The second is the family of questions asked. For the 0.5B model, the placebo's effect is twice as large on one half of our question bank as the other.

Using the same judge to re-judge outputs that are identical we get different decisions 11 times. These decisions range from what answer a model gave, including their base answers without a cue. Using a judge from a different family to rescore the same outputs, it found a different answer on 12-17 percent of the smallest model's outputs versus 3 percent when the same judge is used. We still see the same effect at 0.5B with the placebo causing a high lower bound of influence. We finish with a list of controls that are helpful for future use of this measurement method.

1 Introduction

We measure a model's CoT faithfulness by first prompting the model with a mathematical word problem and recording its baseline response as the base answer. We don't validate the accuracy of this base answer as we only want to measure its faithfulness, not its strength. (For the 0.5B model, the answer is wrong more than half the time). We then prompt again with a cue that suggests a particular wrong answer. This cue can come in the form of a sentence in the system prompt suggesting a source of authority believes a particular answer or be embedded in the user turn.

We then measure two different metrics. First is influence, the fraction of all trials where the answer changed after a cue is introduced and the second is verbalization, the fraction of the trials where the model changed its answer where the model attributes its change to the cue. We exclude "dead"

Preprint.

model	small set, default	small set, matched	large set, default	large set, matched
7B	0.011	0.029	0.009	0.022
3B	0.029	0.017	0.040	0.028
1.5B	0.214	0.054	0.237	0.116
0.5B	0.392	0.214	0.396	0.270

trials, where the base answer is already the cue's target answer. An LLM judge (Sonnet 4.6) then extracts the answer as well as detects whether the reasoning mentions the cue.

In terms of prior work, Turpin et al. and Lanham et al. were the first to look at CoT faithfulness through interventions to the prompt. Chen et al. were the first to measure CoT faithfulness in its modern method, scoring verbalization as a proportion conditional on the model adopting the hint's answer. There has been other work into the validity of this method. Bentham et al. showed that normalising one nuisance term nullifies the finding that small models are less faithful. We also take inspiration from Garcia, who suggested the content-free control of the kind our placebo implements. These have helped suggest the idea that this measurement can be confounded.

We measure how confounded it is in actuality.

2 Setup: Models, Questions and Runs

The LLMs we tested for CoT faithfulness were Qwen 2.5-Instruct at 0.5B, 1.5B, 3B and 7B parameters. We used the same GPU, dtype, temperature (0), one request at a time for all the model generations. We used two types of cues, each with its own placebo/control. The questions we prompted the models with were arithmetic word problems grouped into families that share the same template. Each test resampled based on families as questions from the same family aren't independent.

We used two sets of questions throughout: a small one (87 Qs from 12 template families) and a large one (163 Qs from 24 families), a superset of the small one. Each of the questions were asked to each model under two decoding configurations, the serving default from each model's sampling settings and a matched configuration where we used the same decoding parameters for each model. (See Section 3 for why this matters). We used the same model for the judge throughout.

3 Influence: A Placebo Influences Small Models Almost As Much As Cues

When testing the effect of the placebo, we just changed the prompt. The placebo doesn't mention an answer. The rate of the placebo's influence is the lower bound for a valid influence number.

Comparing the effect of the placebo and the cued influence directly for the 0.5 B model outputs, the placebo produces 85% of the purported cue influence on the matched large-set run (0.270 of 0.319) and 94% for the small-set run (0.214 of 0.229).

The decoding configuration had a rather large impact on the outputs with the placebo. The current vLLM serving default gets the repetition penalty value from the model's configuration file. For Qwen 2.5, there are two different values: 1.05 for 7B and 3B param models and 1.1 for 1.5B and 0.5B param models. This affects the token scores and thus our generations. Using the same decoding configuration for all the models decreases the slope of the lower bound from the placebo from 0.11 to 0.07 on the large set and from 0.11 to 0.05 on the small set of questions. The judge configuration also differed between the default and matched collections, so this comparison bounds the decoding contribution from above rather than isolating it.

The lower bound from the placebo also depends on the family of questions asked. If we only consider the 12 question families in the small set gives the lower bound slope of 0.04, while adding the 12 families in the large set gives 0.10. The 0.5B model lower bound is 0.18 on the first half and 0.36 on the other.

Statistically, fitting a line through four points leaves us 2 degrees of freedom, where significance requires a t above 4.303, so none of the slopes above clears a p value of 0.05 by that test. However, resampling families instead, the lower bound slope is $[+0.039, +0.102]$ on the large set and $[+0.016, +0.097]$ on the small set, both excluding zero.

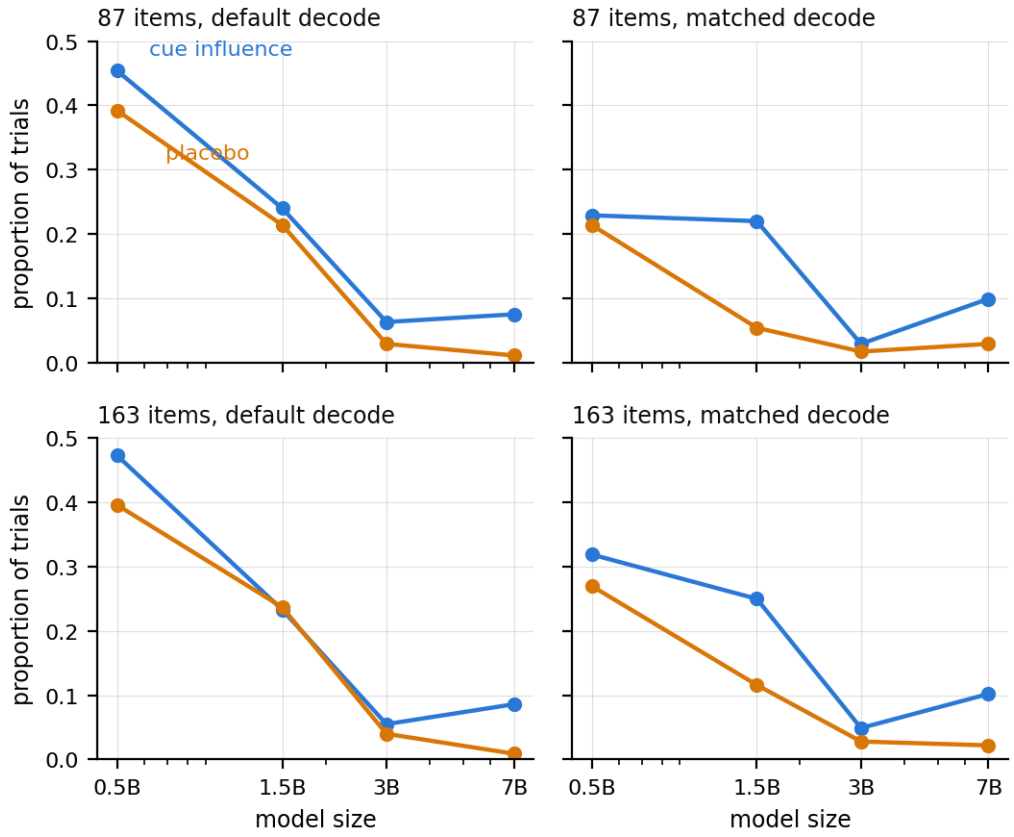

Figure 1: Measured cue influence and placebo influence by model size, per collection condition.

model	small set, default	small set, matched	large set, default	large set, matched
7B	7/13 = 0.54	13/17 = 0.76	22/28 = 0.79	29/33 = 0.88
3B	2/11 = 0.18	1/5 = 0.20	4/18 = 0.22	2/16 = 0.13
1.5B	6/37 = 0.16	1/37 = 0.03	13/70 = 0.19	6/80 = 0.08
0.5B	4/59 = 0.07	0/32 = 0.00	4/123 = 0.03	1/86 = 0.01

4 Verbalization: Almost None when Influence is Largest

The other metric is verbalization: of the trials where the model changed its answer, what fraction of those have reasoning that mentions the cue? The table shows both the counts and the rates.

Looking at the results for the 7B model, remember that the lower bound of influence from the placebo is 1 to 3 percent and thus responses with a different answer are mostly due to the cue. We see that the model here behaves faithfully as the model mentions the cue most of the time, far higher than the lower bound from the placebo. However, the 0.5B model rarely ever mentions the cue.

But using our understanding from the section on influence, this can be explained by other means than unfaithfulness. Because the placebo already has such a great impact on the answer the model provides, it is plausible that the change in answer is not due to the cue, but simply due to random occurrence. Thus, the model possibly has no need to acknowledge the cue so the model is no less faithful as it didn't pick its new answer with the cue information.

model	answer changed	cue mentioned (kappa)	placebo verdict	extracted answer
7B	1.000	1.000 (kappa 1.00, n 17)	1.000	1.000
3B	1.000	1.000 (kappa 1.00, n 5)	1.000	1.000
1.5B	0.994	1.000 (kappa 1.00, n 36)	1.000	0.992
0.5B	0.977	1.000 (kappa undefined, n 25)	0.962	0.973

5 The Judge is Inconsistent with Itself and other models

After running the models on the entirety of the question set, we run the same judge (Sonnet 4.6) on the answers twice. Between these two runs, the same judge extracts a different answer 10 times, leading to 8 real disagreements. 3 placebo answers also changed.

The table here shows the fraction of the decisions where our second pass with the same judge stayed the same.

The verbalization column has 1.0 everywhere, but the underlying effect is different as shown by the kappa between 7B and 0.5B outputs. At 0.5B, every trial was labelled as "did not mention" the cue, so trivially the label stayed the same between judge outputs and the Cohen's kappa is undefined. However, looking at the other decisions the judge is least self-consistent for the smallest model outputs.

Also, the kind of decisions that the judge disagrees with can have very different impacts on the score. 4 of 8 disagreements were about the base answers a model outputs. This impacts whether the later cued trials are discarded and so this greatly affects the denominator. 8 trials the first time we ran the judge counted were discarded for the second run. So, a judge that is 99 percent self-consistent can impact the results by far more than 1 percent.

So far, we have re-used the same judge model and focused on self-consistency instead of correctness. Now, we re-judge the runs with a second judge model from a different family (GPT-4o). Unsurprisingly, disagreement between the judges is several times larger than the disagreement of the judge to itself: for the 0.5B answers the two judges extract a different answer from the exact same text on 12-17 percent of answers, versus 2.7 percent for the same judge re-run. Similar to the self-consistent decisions, choices about the extracted answer, changed answer and placebo answers agree least for 0.5B model outputs. We also see that the number of trials with changed answers changes significantly. At 0.5B whether a trial counts changes for 14-18 trials per run based on the judge model.

For decisions about mentioning the answer, the two judges only agree exactly for the 7B model outputs, with kappa 1.0. At the middle model sizes, kappa falls to below 0.32. The main result stays true regardless of the choice of judge. Scored by gpt-4o, the 0.5B model placebo lower bound is still 83-94 percent of measured influence.

6 Recommended controls

Here are some tips for controls that are helpful as we ran into issues with our own research. Each of these suggestions arises because we got a confident wrong number without it!

Decode every model with the same sampling settings and verify against each checkpoint's config. Run statistics at the level of question families rather than questions and find the smallest p-value that design can produce. Score the outputs with a different judge. Include a positive control, a very powerful cue, so that you can confirm that the model can be influenced by these cues. For cue injection, exclude and keep track of trials where the base answer is already equivalent to the cue's target. Compare models on the same questions.

7 Limitations

We only tested one model family, four sizes and only on arithmetic word problems. We used two judges from different families to score the runs and our results remain consistent with both judges. However, agreement between the judges doesn't necessarily mean certainty in the correctness and

where they disagree we can't tell what is correct. We also didn't run a positive control so at 0.5B, "never verbalises" and "undetectable here" can't be separated.

References

Turpin, M., Michael, J., Perez, E., and Bowman, S. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. arXiv:2305.04388, 2023.

Lanham, T., et al. Measuring Faithfulness in Chain-of-Thought Reasoning. arXiv:2307.13702, 2023.

Chen, Y., et al. Reasoning Models Don't Always Say What They Think. arXiv:2505.05410, 2025.

Bentham, O., Stringham, N., and Marasovic, A. Chain-of-Thought Unfaithfulness as Disguised Accuracy. TMLR, 2024. arXiv:2402.14897.

Garcia. The Last Word Often Wins. arXiv:2605.10799, 2026.