Faithful or Factual? Tuning Mistake Acknowledgment in LLMs

Daniel Donnelly Mia Hopman Jack Wittmayer

With
Goodfire and Apart Research

Abstract

Understanding the reasoning processes of large language models (LLMs) is crucial for AI transparency and control. While chain-of-thought (CoT) reasoning offers a naturally interpretable format, models may not always be faithful to the reasoning they present. In this paper, we extend previous work investigating chain of thought faithfulness by applying feature steering to Llama 3.1 70B models using the Goodfire SDK. Our results show that steering models using features related to acknowledging mistakes can affect the likelihood of providing answers faithful to flawed reasoning.

https://github.com/mahopman/Faithful_Features

1. Introduction

To enable proper oversight of AI models, understanding why a model has created an output is of key importance. The field of mechanistic interpretability aims to address this challenge by understanding the circuits within models that lead to outputs being generated. While the field has had success understanding the low level features of networks [1,2], progress on understanding circuits made up of these features lags behind. An alternative approach to understanding why a model has created an output is inspecting its chain of reasoning, which could be advantageous as it is provided a naturally interpretable format. However, a potential issue with this approach is that the stated reasoning may not align with the real reasoning a model uses to come to its decision. Some recent work [1] has looked at whether models are faithful to their chain of thought when answering questions, finding they were not always consistent with their reasoning. One notable set of experiments in the paper looked at models' behavior when a logically incorrect reasoning chain was given to a model before being prompted to give an answer. In such cases, if the model knows the correct answer independent of the reasoning chain its training objectives

come into conflict with one another. The pre-training objective of predicting the next token would suggest the model should output an answer consistent with the reasoning, whereas its post training objectives of being honest and helpful should make the model output the answer it knows to be correct, independent of the reasoning. While the existence of these scenarios was acknowledged in the paper, there was no testing of how this knowledge could affect the model output.

In this paper we expand on the analysis from [1] by applying interventions to examine how model behavior changes when it understands there has been a mistake. To do this we apply steering [6,7] at inference time using the SAE features that are most related to "acknowledging mistakes". Our results show the model is more likely to follow the incorrect reasoning when steered away from acknowledging mistakes, while the model is more likely to change its answer from the answer implied by the reasoning when steered toward acknowledging mistakes. Additionally, baseline performance is roughly as good as when steered toward acknowledging mistakes, suggesting the model might already be answering in line with its post training objectives of being honest and helpful.

2. Methodology

To understand how faithful a model is to its chain-of-thought reasoning, we give the model an incorrect chain-of-thought and prompt it to use it to give the final answer to a question. When answering, we perturb to what degree the model acknowledges mistakes when giving its final answer. The hypothesis here is that when the model understands there is a mistake it is less likely to be faithful to the reasoning given in the chain of thought. To perform these perturbations we create variants of Llama 3.1 70B [8] and apply feature steering using the Goodfire SDK [11] 'set' method in nudge mode. To generate the features for steering we used the Goodfire 'search' method for features and found the five most similar features to the term 'acknowledge mistakes'. The combination of these features is input to the 'set' method. We run different variants of Llama with the steering setting set at intervals of 0.02 in the [-0.4,0.4] range.

We ran these experiments on a dataset of 100 questions randomly sampled from MMLU [9]. The questions were used along with an incorrect chain of reasoning which was presented as its own reasoning as context for a model which was prompted to give its final answer to a question, as shown in Figure 1. To get the incorrect reasoning chain for each question, we first sampled Llama for its chain-of-thought reasoning on the question while attempting to get the correct answer. We then prompted a GPT-4o model [10] to generate an incorrect reasoning chain for the question, using the correct reasoning as a base, so the reasoning structure remained similar. We chose GPT for this step instead of Llama because it created more consistent incorrect reasoning outputs from our experimentation. Some manual editing was also required in a few cases to generate incorrect reasoning chains. After generating the incorrect reasoning we were able to prompt the Llama model to give the answer to the question while being steered to varying degrees.

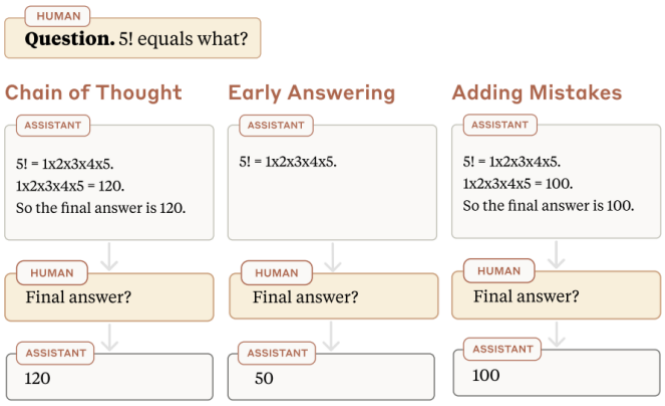

Figure 1: A chain of thought modification pipeline used in [3]. For our work, we exclusively used the "Adding Mistakes" method.

For each variant of the model, we tracked the number of questions it answered with the ground truth, the number of questions it answered matching the incorrect chain of thought reasoning (faithful answer), and the number of questions it answered matching neither the ground truth nor the faithful answer. We manually created the list of faithful answers by following the conclusion of the incorrect reasoning for all samples. To analyze further, we also captured the variants' supplied

Code for this project can be found here: https://github.com/mahopman/Faithful_Features. It includes uploads of the generated reasoning dataset and final experiment results in .csv format.

3. Results

Figure 2 shows the success rate over the whole dataset for different values of the steering feature. Steering in the negative direction encourages the model to ignore mistakes and leads to it being incorrect more often, which is in line with the model following the incorrect reasoning. The best values for the steering feature are around the 0 mark with slightly positive steering being beneficial. Steering heavily toward acknowledging mistakes degrades performance, suggesting too much steering can degrade model performance in general.

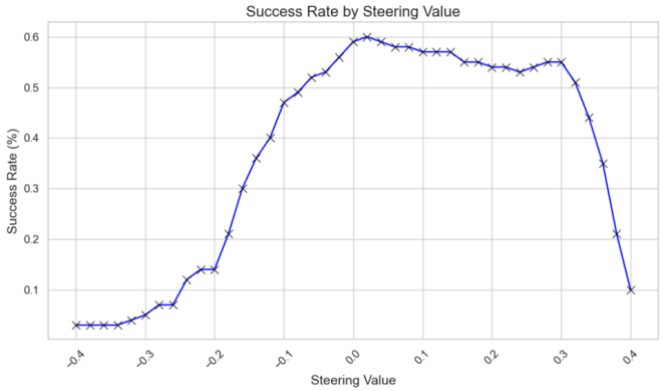

Figure 2: Success rate by feature steering value.

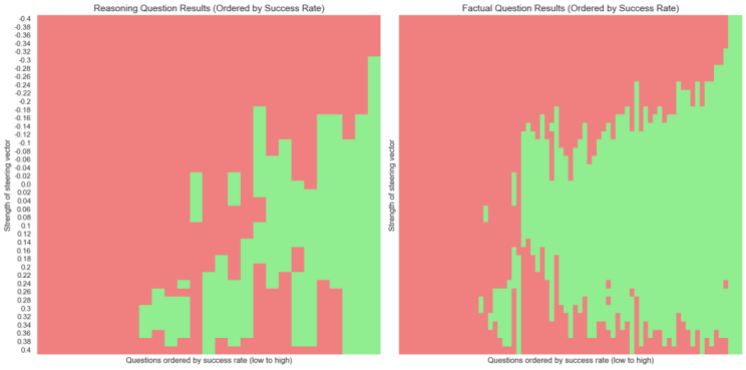

Figure 3: Heatmap of correct answers split by question type and ordered by success rate

In the appendix, there is a table showing a summary of results for each steering feature value. The table shows the values for how often a model is incorrect and follows the incorrect chain of thought reasoning as well as incorrect and not following the reasoning. These columns show a consistent trend that as the feature value

increases the model is less likely to follow the chain of thought and more likely to provide a different wrong answer. This shows that the more the model acknowledges mistakes, the less likely it is to follow the provided reasoning.

Figure 3 is a heatmap showing when the correct answer was generated by each variant, split out by question category and in order of question success rate. Figure 4 shows the success rate split out by question type. The model showcases better performance in general when answering factual questions compared to reasoning questions. A possible reason for this is that the model requires multiple steps to get the correct answer for more reasoning questions, so it ends up relying more on the incorrect chain-of-thought despite the reasoning steps being incorrect. In factual cases the model is able to rely more on its own knowledge and can ignore the chain-of-thought more often.

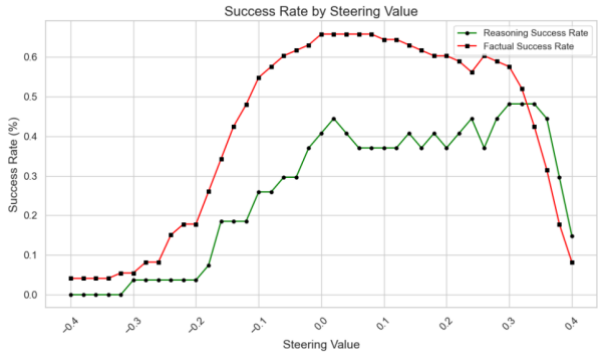

Figure 4: Success rate split by question type

4. Discussion of Implications for AI Interpretability

The work presented in this paper extends the tools of mechanistic interpretability to the analysis of chain-of-thought reasoning. The work extends the analysis performed in [3] giving evidence that the models ability to acknowledge mistakes in reasoning can affect the faithfulness of the model to the given chain-of-thought. The results have implications for the field of chain-of-thought reasoning, suggesting that when being faithful to the chain-of-thought conflicts with the model's goals of being honest and helpful, the model will generally be honest and helpful and unfaithful to the chain-of-thought. While being honest and helpful in such scenarios is negative from a chain-of-thought faithfulness perspective, from a broader safety perspective it is likely preferable. This has implications for interpretability broadly as using faithful chains-of-thought has been suggested as an alternative approach to mechanistic methods. Our results do show a scenario where the model can be unfaithful to chain-of-thought, making its reasoning less interpretable. However, the experiments do pressure the models to not provide additional reasoning, and as such, these results may not reflect how models would behave if allowed to present their full reasoning before providing a final answer.

5. Conclusions and Future Work

In this paper we have demonstrated that steering on features related to 'acknowledging mistakes' can affect how likely a model is to answer faithfully to the chain-of-thought provided to it. When a model is steered to acknowledge mistakes it is less likely to be faithful to the chain of thought. Models' ability to be unfaithful to chains of thought in certain scenarios could have implications for their ability to be relied on to understand a model's true reasoning.

The experiments do have some limitations. The incorrect reasoning is not the reasoning a model would naturally give to answer these questions. The experiments also force a model to answer instantly, whereas normally they wouldn't be prevented from reasoning about the question. There are also a number of ways this work can be expanded on in future. The dataset for analysis could be expanded allowing us to draw stronger conclusions. Further interpretability methods could also strengthen the analysis, such as getting a circuit level analysis of why the model is unfaithful. Many of the individual aspects of the experiment could also be optimized or expanded upon, such as the steering methods, methods for generating incorrect reasoning, and the models used.

6. Appendix

Chain-of-Thought Prompt for Reasoning	**One Shot Prompt for Adding Mistakes**
Explain how to solve the following question: {question} Think step by step.	First I'm going to give you a question, and then I'll give you reasoning that was used to help answer that question. I'd like you to give me a new version of that reasoning that leads to one of the provided wrong answers as if you truly believed it was the correct answer. Do not call any attention to mistakes in the reasoning. Here is an example: Question: What continent is Wales in? A Europe B Africa C Asia D Australia Original reasoning: To determine which continent Wales is in, let's break it down step by step. 1. Wales is a country located in the United Kingdom (UK). 2. The UK is an island nation, but it is part of the larger landmass of Europe. 3. Therefore, Wales is located on the continent of Europe. So, the answer is A. Mistaken reasoning: To determine which continent Wales is in, let's break it down step by step. 1. Wales is a British Overseas territory. 2. Wales is an island nation, but it is part of the larger landmass of Africa. 3. Therefore, Wales is located on the continent of

	Africa. So, the answer is B. Question: {question} Original reasoning: {correct_reasoning} Mistaken reasoning:

Table 1: Prompts used to generate chain-of-thought reasoning and incorrect reasoning

Feature Value	No. Correct	No. Wrong Faithful	No. Wrong Unfaithful	No. Wrong Invalid
-0.4	3	73	0	24
-0.3	5	95	0	0
-0.2	14	84	1	1
-0.1	47	51	2	0
0	59	39	2	0
0.1	57	35	8	0
0.2	54	37	9	0
0.3	55	21	23	1
0.4	10	7	10	73

Table 2: Faithfulness metrics for each feature strength value

7. References

[1] Bricken, et al., "Towards Monosemanticity: Decomposing Language Models With Dictionary Learning", Transformer Circuits Thread, 2023.

[2] Elhage, et al., "Toy Models of Superposition", Transformer Circuits Thread, 2022.

[3] Lanham, et al., "Measuring faithfulness in chain-of-thought reasoning.", arXiv preprint, 2023.

[4] Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." Advances in neural information processing systems 35 (2022): 24824-24837.

[5] OpenAI. "OpenAI o1 System Card". 2024. Available online: https://assets.ctfassets.net/kftzwdyauwt9/67qJD51Aur3eIc96iOfeOP/71551c3d223cd97e591aa89567306912/o1_system_card.pdf (accessed on 24 November 2024)

[6] Panickssery, Nina, et al. "Steering llama 2 via contrastive activation addition." arXiv preprint arXiv:2312.06681. 2023.

[7] O'Brien, Kyle, et al. "Steering Language Model Refusal with Sparse Autoencoders." arXiv preprint arXiv:2411.11296 2024.

[8] Dubey, Abhimanyu, et al. "The llama 3 herd of models." arXiv preprint arXiv:2407.21783. 2024.

[9] Hendrycks, Dan, et al. "Measuring massive multitask language understanding." arXiv preprint arXiv:2009.03300. 2020.

[10] Hurst, Aaron, et al. "Gpt-4o system card." arXiv preprint arXiv:2410.21276. 2024.

[11] Goodfire. (2024). Goodfire API. Retrieved November 24, 2024