Encouraging Chain-of-Thought Reasoning in Language Models Through Feature Steering

Soumyadeep Bose Kutay Buyruk Shreyans Jain Thomas Walker

With
Goodfire and Apart Research

Abstract

Large language models can improve their question-answering abilities through chain-of-thought reasoning, but current methods rely heavily on engineered prompts which lack reliability and theoretical foundation. In this work, we propose using feature steering as a more systematic approach to encourage chain-of-thought reasoning. Rather than crafting specific prompts, we directly modify the model's internal mechanisms by identifying and activating features associated with chain-of-thought reasoning. Through analysis of model activations across direct and reasoned responses on a curated dataset, we identify key features that characterize chain-of-thought reasoning at both the token and full-prompt level. Our experimental results on the MathQA dataset demonstrate that feature steering can successfully elicit more accurate chain-of-thought reasoning compared to baseline responses.

[Full code repository here]

Keywords: Model Reasoning, Feature Steering

1. Introduction

Large language models contain a large amount of information, however, eliciting the model to generate this information reliably and effectively has been an intense topic of research. A particularly effective strategy has been to encourage the models to perform *chain-of-thought* reasoning (Wei *et al.*, 2023). Consequently, there have been numerous attempts to try and elicit this behavior in large language models, through prompting pipelines (Kojima *et al.*, 2023) or fine-tuning on chain-of-thought data (Chung *et al.*, 2022) or modifying the text generation algorithm (Wang and Zhou, 2024) which helps get an accurate representations of the models' ability.

Recent work has applied interpretability techniques to try and understand this phenomena. For instance, (Dutta *et al.*, 2024) uses mechanistic interpretability to elucidate the mechanisms involved in the carrying out chain-of-thought reasoning. Similarly, (Wu *et al.*, 2023) uses feature attribution methods to understand the internal processes of the model during chain-of-thought reasoning. Here we take this analysis one step further, by identifying features that are prominent in chain-of-thought reasoning. Our motivation is to derive more principled methods for inducing chain-of-thought reasoning that is not reliant on heavy prompt engineering. On the one hand, this would allow us to improve, or suppress, model performance, giving more control to deployers on the potential implications their models could have. By not having to rely on prompt engineering to elicit chain-of-thought reasoning, we can better evaluate the inherent capabilities of the models, rather than having to account for biases introduced in the prompts. Which also helps us better interpret what the models are thinking. Likewise, we can elicit chain-of-thought reasoning when we do not have prior chain-of-thought reasoning steps to few-shot the model.

2. Overview

2.1 Dataset Curation

We started by collecting examples of the Llama-3-8B-Instruct model answering questions directly and using chain-of-thought reasoning, through few-shot prompting. In total we collected 20 responses on questions, where for each question we got direct and reasoned. In section 4.1 there is an example of such an exchange we recorded with the model.

We used this dataset to conduct feature analysis from which we could then derive our steering methods. To then test these steering methods we collected samples of math questions from MathQA (Amini *et al.*, 2019) and MATH (Hendrycks *et al.*, 2021). Examples of these questions can be found in section 4.1.

2.2 Feature Analysis

With our initial set of examples we set about analyzing the features that were active during these responses, with the goal of being able to distinguish the responses given directly and those given using chain-of-thought reasoning.

2.2.1 Next Token Analysis

Due to the autoregressive nature of language models, it seems reasonable that optimizing our feature steering to match the feature activation at the first token of a chain-of-thought reasoning sequence should be significant to control this behavior in the model. Therefore, we explore the differences between the feature activations at the first token of the model responses in our dataset. More specifically, we take the first token of the responses and identify the top eight features that activate on that token.

From this we make the general observation that the model is more confident and enthusiastic in its answer when it is about to respond using chain-of-thought reasoning. We deduce this from the first two tables in section 4.2.1, which show that the most frequently activating features for the chain-of-thought reasoning answers are those expressing enthusiasm and initiating a step-by-step explanation. Whereas, those most frequently firing for direct responses are those expressing uncertainty, limitations in the capabilities of the model or seeking clarification.

This is further evidenced by ranking the features by how intensively they fire across the responses, shown in the second two tables of section 4.2.1. What we observe is that the feature most active in the chain-of-thought response is that of taking action, or preparing to undertake reasoning, whereas the most active feature for direct responses is related to seeking clarification.

2.2.2 Full Prompt Analysis

In addition to the first token, we analyzed the features for the complete prompt as well. We used contrastive search to filter out features which can be used to initiate chain-of-thought reasoning. Internally, contrastive search calculates the differences between the nonzero mean feature activations and outputs best features to use for steering towards chain-of-thought reasoning.

We observe that the model is using polite and appreciative language structure for communicating when engaging with chain-of-thought reasoning. Along with this, next frequently activating features are the ones initiating a step-by-step explanation, which matches with our previous findings in section 2.2.1.

2.3 Feature Steering

In these sections we leverage the findings of section 2.2 to effectively steer the model to providing responses utilizing chain-of-thought reasoning. We aim for effective chain-of-thought reasoning, where by effective we mean succinct reasoning that leads to the correct answer to the questions.

2.3.1 From Next Token Analysis

Using the insights of section 2.2.1, we develop and test steering methods on Llama-3.1-70B-Instruct by prompting the model zero-shot with a subset of questions from the MathQA dataset. From preliminary experiments, the results of which we discuss in section 4.2.2, we found that the most effective strategy for eliciting effective chain-of-thought reasoning was to steer the model with a combination of the following features.

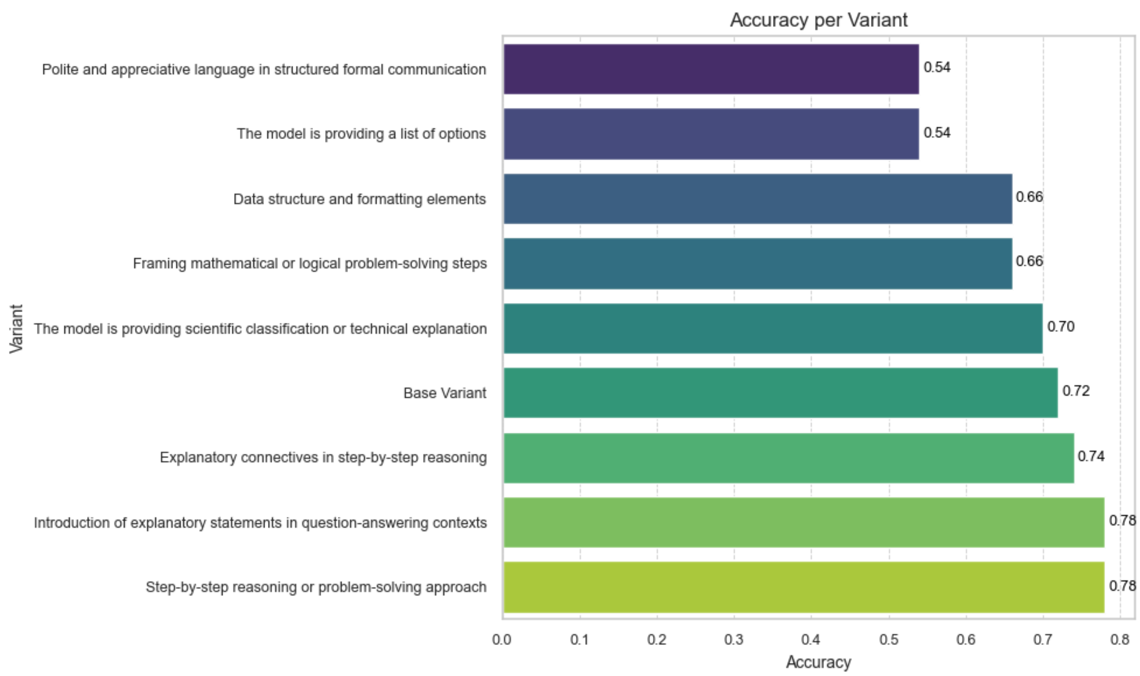
1. “Mathematical variables and numerical values in formal context”
2. “The assistant’s turn to speak”
3. “The assistant is starting a step-by-step explanation”

Our evaluating dataset contains 24 questions with verifiable answers, from the MATH dataset. We manually check whether the model answered the question correctly in its response, and we also record the length of the response given by the model. We use the response length, in terms of the number of characters, as a proxy for coherency. Our results are summarized in the table of section 4.2.3, from which we deduce that feature steering can trigger more concise chain-of-thought reasoning, that can even allow the model to answer questions correctly that it previously got wrong.

2.3.2 From Full Prompt Analysis

Drawing on our observations from Section 2.2.2, we derive a list of 8 features along with their thresholds (Section 4.2.4 in Appendix) which gave us good preliminary results on chain-of-thought reasoning. We test our steering methods by zero-shot prompting Llama-3.1-70B-Instruct on a sample of 50 questions from the MathQA dataset.

From our experiments, we found that few of the features are very effective in eliciting chain-of-thought reasoning, beating the baseline model variant in accuracy. While we weren't able to get an increase in model performance for all the feature steering variants, these results give us optimism that post a deeper analysis, feature steering can be an effective technique to elicit chain-of-thought reasoning and improve performance on various tasks.



3. Discussion and Conclusion

In this project we set out to understand the features of chain-of-thought reasoning. We found surprising features labeled as being parts of chain-of-thought sequences including those linked to and politeness. These features were subsequently effective at eliciting effective chain-of-thought reasoning. With more time we would like to perform a more fine-grained analysis of the features, and observe how they are affected by factors such as the question type and topic. We would also like to verify our results on larger datasets.

With this work we believe we have made the first efforts towards achieving our initial goals. Namely, understanding chain-of-thought reasoning at a feature level, and determining if we could use feature steering to elicit this form of reasoning without the need of extensive prompt engineering. In the hope of providing techniques for researchers to better control and understand the capabilities of these models, as well as get an unbiased perspective on what the model is thinking.

References

Amini, A. *et al.* (2019) 'MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms'. arXiv. Available at: https://doi.org/10.48550/arXiv.1905.13319.

Chung, H.W. *et al.* (2022) 'Scaling Instruction-Finetuned Language Models'. arXiv. Available at: https://doi.org/10.48550/arXiv.2210.11416.

Dutta, S. *et al.* (2024) 'How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning'. arXiv. Available at: https://doi.org/10.48550/arXiv.2402.18312.

Hendrycks, D. *et al.* (2021) 'Measuring Mathematical Problem Solving With the MATH Dataset'. arXiv. Available at: https://doi.org/10.48550/arXiv.2103.03874.

Kojima, T. *et al.* (2023) 'Large Language Models are Zero-Shot Reasoners'. arXiv. Available at: https://doi.org/10.48550/arXiv.2205.11916.

Wang, X. and Zhou, D. (2024) 'Chain-of-Thought Reasoning Without Prompting'. arXiv. Available at: https://doi.org/10.48550/arXiv.2402.10200.

Wei, J. *et al.* (2023) 'Chain-of-Thought Prompting Elicits Reasoning in Large Language Models'. arXiv. Available at: https://doi.org/10.48550/arXiv.2201.11903.

Wu, S. *et al.* (2023) 'Analyzing Chain-of-Thought Prompting in Large Language Models via Gradient-based Feature Attributions'. arXiv. Available at: https://doi.org/10.48550/arXiv.2307.13339.

4. Appendix

4.1 Dataset Curation

Below is an example exchange we had with the model to get direct and chain-of-thought responses from it, from which we could then conduct feature analysis.

<u>*Answering Directly:*</u>

User

Q: Which is a faster way to get home?
Option 1: Take an 10 minutes bus, then an 40 minute bus, and finally a 10 minute train.
Option 2: Take a 90 minutes train, then a 45 minute bike ride, and finally a 10 minute bus.

A: Option 1
Q: Which is a faster way to get to work?
Option 1: Take a 1000 minute bus, then a half hour train, and finally a 10 minute bike ride.
Option 2: Take an 800 minute bus, then an hour train, and finally a 30 minute bike ride.

A:

Model

I'd say Option 2 is the faster way to get to work!

<u>Answering with Chain-of-Thought Reasoning:</u>

User

Q: Which is a faster way to get home?
Option 1: Take an 10 minutes bus, then an 40 minute bus, and finally a 10 minute train.
Option 2: Take a 90 minutes train, then a 45 minute bike ride, and finally a 10 minute bus.

A: Option 1 will take 10+40+10 = 60 minutes. Option 2 will take 90+45+10=145 minutes.Since Option 1 takes 60 minutes and Option 2 takes 145 minutes, Option 1 is faster.
Q: Which is a faster way to get to work?
Option 1: Take a 1000 minute bus, then a half hour train, and finally a 10 minute bike ride.
Option 2: Take an 800 minute bus, then an hour train, and finally a 30 minute bike ride.

A:

Model

Let's calculate the total time for each option!
Option 1: 1000 minutes (bus) + 30 minutes (train) + 10 minutes (bike) = 1040 minutes
Option 2: 800 minutes (bus) + 60 minutes (train) + 30 minutes (bike) = 890 minutes
Since Option 2 takes 890 minutes and Option 1 takes 1040 minutes, Option 2 is the faster way to get to work!!

Below is an example of a question from the MathQA dataset.

Q: Sophia finished ⅔ of a book. She calculated that she finished 90 more pages than she has yet to read. How long is her book?

Below is an example of a question from the MATH dataset.

Q: Find the point on the line $y = -3x + 5$ *that is closest to the point* $(-4, -2)$.

4.2 Feature Analysis

4.2.1 Next Token Feature Analysis

Chain-of-Thought Responses		
Feature Index	Feature Description	Frequency
9430	The model initiating correct answer	9
42860	The model's turn to speak, especially with enthusiasm	8
16397	The model initiating step-by-step explanation	7
28805	The model is considering or preparing to take an action	6
22207	Introduction of variables or assumptions in mathematical reasoning	6

64369	The model is beginning a step-by-step explanation	6
36354	The model needs to express uncertainty or limitations	6
2531	The assistant needs to explain its limitations or refuse inappropriate requests	6

Direct Responses		
Feature Index	Feature Description	Frequency
9430	The model initiating correct answer	9
36354	The model needs to express uncertainty or limitations	5
2531	The assistant needs to explain its limitations or refuse inappropriate requests	5
27740	The model needs clarification	5
53239	Definite article 'the' indicating specificity or known reference	5

Chain-of-Thought Responses		
Feature Index	Feature Description	Cumulative Activation
28805	The model is considering or preparing to take an action	25.20
22207	Introduction of variables or assumptions in mathematical reasoning	17.54
36354	The model needs to express uncertainty or limitations	15.65

Direct Responses		
Feature Index	Feature Description	Cumulative Activation
36354	The model needs to express uncertainty or limitations	12.33
53249	Definite article 'the' indicating specificity or known	9.90

Direct Responses		
	reference	
9430	The model initiating correct answer	8.63

One thing we also observed is the features tend to fire more intensely, and across a smaller number of features in the case of chain-of-thought response. We quantified this by calculating the skewness of the feature activations across the top eight firing features. The chain-of-thought response had a mean skewness of 1.4, whereas the direct responses had a skewness of 0.79, indicating that the chain-of-thought features fired more intensely on a fewer number of features.

4.2.2 Next Token Feature Steering Exploration

To understand which steering techniques were viable, we first conducted some experiments on our initial set of 20 examples on which we conducted the feature analysis.

In general, what we discovered was that pinning was more effective at eliciting the desired behavior compared to nudging. Moreover, pinning at a value of `0.6` provided the correct balance of behavior.

Interestingly, steering toward features with labels containing "step-by-step" was not too beneficial, as the model would just end up repeating the word "Let". In fact, steering toward enthusiasm gave better results.

Since this example dataset contains examples of the model answering questions directly, we found that steering toward enthusiasm made the model more likely to respond through chain-of-thought. However, some of these chain-of-thought reasonings didn't lead to the correct answers, this was a problem we encountered when we then scaled the approaches.

During this exploratory phase we recorded when the feature steering elicited chain-of-thought behavior when the model was prompted to give direct responses, and whether these led to the model giving the correct answer. Moreover, we recorded what the feature steering behavior had on responses that were already prompted to give answers using chain-of-thought reasoning.

Baseline				Step-by-Step nudging				Enthusiasm nudging				Introducing variables nudging			
CoT Prompting		Direct Prompting		CoT Prompting		Direct Prompting		CoT Prompting		Direct Prompting		CoT Prompting		Direct Prompting	
CoT Reasoning	Correct	CoT Reasoning	Correct	CoT Reasoning	Correct	CoT Reasoning	Correct	CoT Reasoning	Correct	CoT Reasoning	Correct	CoT Reasoning	Correct	CoT Reasoning	Correct
y	y	n	y	y	n	y	n	y	y	y	n	y	n	y	y
y	y	y	y	y	y	y	y	y	y	y	n	y	n	y	y
y	y	n	n	y	y	n	n	y	y	y	y	y	y	n	n
y	y	y	y	y	y	y	n	y	n	y	y	y	y	y	n
n	y	n	y	n	y	n	y	n	y	n	y	n	y	n	y
y	n	n	n	y	y	n	y	y	n	y	y	y	y	y	y
y	n	n	y	y	y	n	y	y	y	y	y	y	y	y	y
y	y	n	y	n	n	n	n	y	n	y	y	y	y	n	y
y	n	y	n	y	y	y	y	y	y	y	n	y	y	y	y
y	y	n	y	y	y	y	y	y	y	y	y	y	y	y	y
y	y	y	y	y	y	y	n	y	y	y	y	y	y	y	y
y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
y	y	n	y	y	y	n	y	y	y	y	y	y	y	n	y
y	y	n	y	n	y	n	y	y	y	y	y	y	y	n	y
y	y	n	y	n	y	n	y	y	y	n	y	y	y	n	y
y	y	n	y	y	y	n	y	y	y	n	y	y	y	n	y
n	y	n	y	n	y	n	y	n	y	y	y	n	y	n	y
y	y	n	y	n	y	n	y	y	y	y	y	y	y	y	y
y	y	n	y	n	y	y	y	n	y	n	y	n	y	y	y
18	17	6	17	13	18	9	15	17	17	16	17	17	18	12	18

Enthusiasm pinning				Step-by-Step pinning				Introducing variables pining			
CoT Prompting		Direct Prompting		CoT Prompting		Direct Prompting		CoT Prompting		Direct Prompting	
CoT Reasoning	Correct	CoT Reasoning	Correct	CoT Reasoning	Correct	CoT Reasoning	Correct	CoT Reasoning	Correct	CoT Reasoning	Correct
y	y	y	n	y	y	y	n	y	n	y	n
y	y	y	n	y	y	y	y	y	n	y	y
y	y	y	n	y	y	n	n	y	n	n	n
y	n	y	y	y	y	y	y	y	y	y	n
n	y	n	y	n	y	n	y	n	y	n	y
y	n	y	y	y	y	n	y	y	y	y	y
y	n	y	y	y	y	y	y	y	y	y	y
y	n	y	y	n	n	n	n	y	y	n	y
y	y	y	n	y	y	y	n	y	y	y	y
y	y	y	n	y	y	y	y	y	y	y	y
y	y	y	y	y	y	y	n	y	y	y	y
y	y	y	y	y	y	y	y	y	y	y	y
y	y	y	y	y	y	y	y	y	y	y	y
y	y	y	y	y	y	n	y	y	y	n	y
y	y	n	y	n	y	n	y	y	y	n	y
y	y	n	y	n	y	n	y	y	y	n	y
y	y	n	y	y	y	n	y	y	y	n	y
y	y	y	y	n	y	y	y	n	y	n	y
y	y	y	y	n	y	n	y	y	y	y	y
y	y	y	y	n	y	y	y	n	y	y	y
19	16	16	15	13	19	11	15	17	17	12	17

4.2.3 From Next Token Analysis Feature Steering Full Results

Below are the full results we obtained when steering the model according to the analysis we did on the first token of the response from the model.

Baseline		1,2,3 Steering		1,3 Steering		2,3 Steering		1,2 Steering	
Correct	Length	Correct	Length	Correct	Length	Correct	Length	Correct	Length
yes	**1181**	yes	1530	yes	1247	no	1263	no	1651
yes	**221**	yes	228	yes	414	yes	244	yes	170
yes	713	yes	495	**yes**	**465**	yes	684	yes	504

yes	776	**yes**	**384**	yes	452	yes	857	yes	561
yes	**682**	yes	1043	yes	793	no	1235	no	928
yes	619	**yes**	**530**	yes	769	yes	856	yes	536
yes	603	yes	507	**yes**	**387**	yes	649	yes	420
yes	**299**	yes	363	yes	320	yes	356	yes	353
yes	268	**yes**	**244**	yes	254	yes	356	yes	247
yes	**142**	**yes**	**142**	yes	301	yes	200	yes	213
yes	204	yes	204	**yes**	**137**	yes	398	yes	270
yes	957	yes	961	yes	840	**yes**	**812**	no	942
no	700	**yes**	**668**	yes	1045	yes	754	no	879
no	1120	no	1129	no	917	no	1108	no	927
yes	**441**	yes	591	yes	640	yes	798	yes	636
yes	461	yes	635	**yes**	**414**	yes	948	yes	495
yes	761	yes	1202	no	951	yes	778	**yes**	**561**
yes	597	yes	692	**yes**	**575**	yes	1239	yes	812
yes	**1100**	no	1020	no	896	no	1189	no	1166
yes	645	yes	683	no	917	**yes**	**542**	**yes**	**542**
yes	976	yes	855	yes	988	**yes**	**742**	**yes**	**742**
yes	**329**	yes	394	yes	399	no	396	yes	396
yes	142	yes	112	**yes**	**102**	yes	142	yes	142
yes	**307**	yes	435	yes	501	yes	481	yes	481

Each row in the table corresponds to a model response on a particular question in our dataset. The highlighted cells are the cells in which the model answers correctly in the shortest length. Of note is the cell we have highlighted in green, which shows an instance where feature steering results in the model getting an answer correct which it previously got wrong.

4.2.4 Full Prompt Feature Steering Analysis:

List of Steering Features and their respective thresholds :

1. "Polite and appreciative language in structured formal communication", 0.5
2. "Data structure and formatting elements", 0.3
3. "Introduction of explanatory statements in question-answering contexts", 0.3
4. "Step-by-step reasoning or problem-solving approach", 0.4
5. "The model is providing a list of options", 0.5

6. "Explanatory connectives in step-by-step reasoning", 0.2
7. "Framing mathematical or logical problem-solving steps", 0,2
8. "The model is providing scientific classification or technical explanation", 0.1

4.3 Code

4.3.1 Feature Analysis

To extract the features given by the assistant on our examples, we would first run the text into the model as

```
context=client.features.inspect([{“role”:“user”,“content”:user},{“role”:“assistant”,“content”:assistant}],model=variant)
```

where `user` and `assistant` were variables containing the sample text. For analysis at the first token given by the assistant we would inspect the context at the token index equal to that of the first token given by the assistant and then apply the `vector()` method to this to identify these features along with their activations.

For analyzing the features at the prompt level we would use the contrastive search method from Goodfire API, `client.features.contrast`. This method helps us filter out features which can be used to steer from one dataset to another (For ex: non CoT to CoT Reasoning).

4.3.2 Feature Steering

To steer the model with the features we identified from analyzing the next token, we used the `set()` method of the model with the features of interest along with `mode=“pin”` and `value=0.6`. At inference time we would run the model using `client.chat.completions.create` with `stream=True`, `max_completion_tokens=512`, `top_p=0`, `temperature=0.`

The features found during next token analysis were used to run feature steering benchmarks on a subset of the MathQA (Amini *et al.*, 2019) dataset. Each feature was steered on a separate variant using the `set()` method with `mode=“nudge”` and varying values. Each variant answered the same set of questions from the dataset, and were instructed to finish their response in the form of “Response: [a|b|c|d]”. A regex filter parsed and extracted possible answers following the format from the model response, which was used to compare against reference answers and measure the accuracy of different variants.

4.4 Explanation of Results

Here we just want to try and reason about some of the results that we found. Consistently, we found that chain-of-thought reasoning activated enthusiasm and politeness features in the models, whereas direct answers came across with more uncertainty.

From a human perspective, this makes some sense. We are often more confident in our answers when we can detail the reasoning behind them. Similarly, if we can reason through an answer, we are often quite excited to share this with other people.

From the training perspective, this phenomenon is also probably not that surprising. After all, these models are fine-tuned using human feedback, and often responses exhibiting chain-of-thought reasoning are perceived as more useful than those just giving direct answers. In these cases the model is trying to be helpful, polite and enthusiastic as it is an assistant. Therefore, it is not surprising that when it is conducting chain-of-thought reasoning it is triggering these types of features.