Probing Internal LLM Activations for Mathematical Reasoning

Sewoong Lee, Isha Chaudhary, Vishal Bharadwaj,
Vinay Purushotham and **Sai Krishna Rohith Kattamuri**
Siebel School Computing and Data Science
University of Illinois Urbana-Champaign
{samuel27, isha4, vishalb3, vp32, skatt24}@illinois.edu

Abstract

One promising approach for *understanding LLMs* is analyzing their internal neuron activations. Although humans cannot interpret the model's weights, parameters, or activations in their raw format, we can install a simple classifier, also known as a *probe*, to determine whether a certain representation resides in a particular component of the model. Recently, there has been research that claimed that we can not only detect a model's internal representation but also control to express a certain type of emotion. This recent research raises questions about whether there is a probe that can observe the accuracy of mathematical reasoning, one of the current weaknesses of LLMs. In this paper, we classify models' incorrect reasoning based on model's internal activations on GSM8k math reasoning task. To the best of our knowledge, our study is the first to probe the model's representations for math reasoning and to verify the feasibility of performance improvement through representation engineering. Various probes applied to different components within the LLM consistently achieved an F1 score of 0.7 in detecting incorrect reasoning on-the-fly, but the classification for general cases was impossible. However, counterintuitively, long answers was more a feature of poor reasoning rather than good reasoning in that the model struggles to reach an answer. Our code and dataset are available at: `https://github.com/SewoongLee/probing-reasoning/`

1 Introduction

Large Language Models (LLMs), while typically exhibiting high performance, are black-boxes and therefore inherently uninterpretable. Various techniques have been developed to *explain* their behaviors (Arrieta et al., 2019; Doshi-Velez and Kim, 2017), which can be broadly classified into post-hoc and inherent interpretability of the model. Post-hoc explanation methods (Lundberg and Lee, 2017; Ribeiro et al., 2016) explain the predictions of models after they are trained normally. Inherent interpretability of the models involves using more interpretable architectures (Riegel et al., 2020) and can be brought about with specialized training, which can harm performance. In this work, we focus on post-hoc interpretability using model probing (Hewitt and Manning, 2019).

LLMs excel at many tasks but are still weak at reasoning, as seen in mathematical datasets like GSM8k (Cobbe et al., 2021). It is both surprising and disappointing that LLMs which are pre-trained on indiscriminate corpus are able to perform math but yet remain clumsy, even at grade school level mathematics.

To understand those emergent capabilities of LLMs, *representation engineering* is often used for this purpose. Burns et al. (2022) suggested that inner representations of correct answers exist, even when the model's final output is wrong. Building on this, Zou et al. (2023) developed methods for visualizing and controlling various inner representations, such as honesty.

Natural questions arise in this context, such as (1) 'what can we observe inside the model when mathematical reasoning goes wrong?' or (2)'If probing for incorrect reasoning is possible, could we control towards the opposite direction to improve the model's reasoning accuracy?' The contributions of this paper are:

- We propose novel math reasoning datasets for contrastive learning, extending GSM8k (Cobbe et al., 2021) with examples of incorrectly generated reasoning.

- We verified previous methods of representation engineering (Zou et al., 2023), which used contrastive datasets and learning to control oppositional representation, demonstrating that mathematical correctness cannot easily be achieved by steering the model towards a more correct direction.

- We demonstrate that we can partly classify the current state-of-the-art LLMs' internal representation of incorrect reasoning with math problem solving tasks.
- We demonstrated that despite various analyses, it is difficult to clearly classify inaccurate reasoning using only the information within a single model, making it challenging to improve reasoning performance based on this.

2 Related Work

2.1 LLM Math Reasoning

It is surprising that even lightweight LLMs (with fewer than 10 billion parameters) can generalize and solve grade-school math problems to some extent, yet disappointing that they often fail to produce correct answers for even the simplest problems. Due to this intriguing aspect, we explore the internal properties of lightweight text-only models. Along with this intriguing nature, math reasoning tasks have been a good testbed for LLMs because (1) there are clear correct answers, and (2) there can be infinitely many combinations of problem variations to assess generalization. Recently, for the efforts to isolate their generalization performance from the effect of data contamination, GSM-Symbolic (Mirzadeh et al., 2024) showed that changing the sub-sentences of the math problem can significantly undermine the performance of some LLMs. Specifically, among the lightweight text-only LLMs, Llama3 (Dubey et al., 2024) showed robust peformance with certain changes, but the model like Mystral v0.1 (Jiang et al., 2023) showed significant performance drop.

2.2 Probe

A probe is a simple model used to evaluate the existence of representations learned by a neural network. Two commonly used probe classifiers are logistic regression and support vector machines (SVM).

Logistic Regression: Logistic regression models the probability of a class $y \in \{0, 1\}$ given an input feature vector $\mathbf{x}$ using the sigmoid function:

$$P(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b),$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$, and $\mathbf{w}, b$ are learned parameters.

SVM with RBF Kernel: Support Vector Machines with a radial basis function (RBF) kernel use a decision function of the form:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b,$$

where $K(\mathbf{x}_i, \mathbf{x}) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}\|^2)$ is the RBF kernel, and α_i, γ, b are learned parameters, assigning $y \in \{-1, 1\}$ based on the sign of $f(\mathbf{x})$. The RBF kernel is chosen for its ability to model complex, non-linear decision boundaries with fewer hyperparameters compared to polynomial kernels.

2.3 Probing and Controlling LLM

Probing, installing a simple classifier to a machine learning model, is a technique that allows us to detect the presence of a model's internal representation when performing a specific task on-the-fly. For example, one notable work of using probing to demonstrate that the model possesses a certain type of information utilized the game of Othello (Li et al., 2023). When game histories are given in sequential text, the fact that simple multi-layer perceptrons (MLPs) can reconstruct the board state indicates that the model is internally using the two-dimensional board information.

One notable approach to probing LLMs is *representation engineering*, suggested by (Zou et al., 2023). While the study focused on engineering emotions or honesty, our work extends this approach to the problem of correct mathematical reasoning. To draw an analogy, this is similar to probe the emotions a human feels when they make a mistake on a math problem. Specifically, we define a set of reasoning tasks $\mathcal{T}$, where $\mathcal{C} \subset \mathcal{T}$ is the set of correctly classified tasks and $\mathcal{I} = \mathcal{T} \setminus \mathcal{C}$ is the set of incorrectly classified tasks. For the model weights W and the task $t \in \mathcal{T}$, we extract sets of neural network activation as follows:

$$A_{\text{correct}} = \{\text{Rep}(W, t)[-1] \mid t \in \mathcal{C}\}$$

$$A_{\text{incorrect}} = \{\text{Rep}(W, t)[-1] \mid t \in \mathcal{I}\}.$$

where Rep_i is a trainable probing functions of all layers at the i^{th} token position to classify weights based on their representations. We then can modify the model by incorporating the probing results:

$$W' = W + \Delta W,$$

where ΔW represents the weight adjustment based on the contrastively learned LoRA with $\mathcal{L} =$

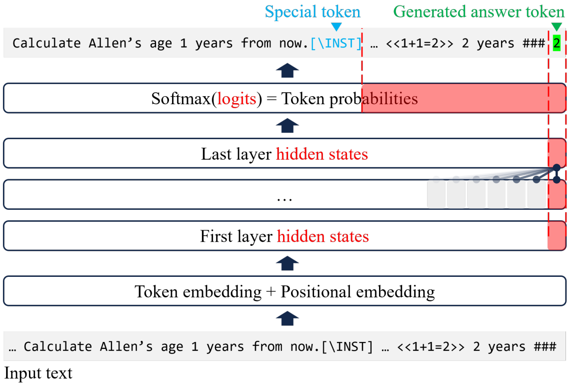

Special token
Generated answer token
Calculate Allen’s age 1 years from now.[\INST] … <<1+1=2>> 2 years ### 2
Softmax(logits) = Token probabilities
Last layer hidden states
…
First layer hidden states
Token embedding + Positional embedding
… Calculate Allen’s age 1 years from now.[\INST] … <<1+1=2>> 2 years ###
Input text

Figure 1: The components of the model we probed in our research. We primarily applied probing to the content within the red-highlighted boxes. We also visualized these components using PCA or histograms, or performed feature engineering by defining custom metrics. The last token or layer is commonly used for probing due to the compressed contextual information it contains.

$\|\mathrm{Rep}(W, t^{+}) - \mathrm{Rep}(W, t^{-})\|_2$, with $\mathcal{L}$ being the loss function, t^{+} is the correct task and t^{-} is the incorrect task.

Prior works (Antverg et al., 2022; Li et al., 2024; Zou et al., 2023; Han et al., 2024) have demonstrated the utility of insights from probing explanations in improving the performance of LLMs during inference. Specifically, these approaches edit the model representations during inference to steer them towards desirable answers. Prior works study model properties such as generation style, truthfulness, memorization, etc., unlike our work, which is focused on mathematical reasoning.

3 Experiments

Llama 3.2 3B demonstrates one of the best performances on mathematical reasoning tasks among light-weight models with less than 10B parameters and also exhibited the robustness to data contamination issues as shown in Mirzadeh et al. (2024). Therefore, as the backbone model, we use Llama 3.2 3B unless otherwise specified.

3.1 Probing for Math Reasoning

When humans solve math problems, sometimes they can feel that they are solving the problems in the wrong way. Will the machine possesses corresponding information within the model? To formalize as a research question: *Can we probe internal representation of incorrect answer?*

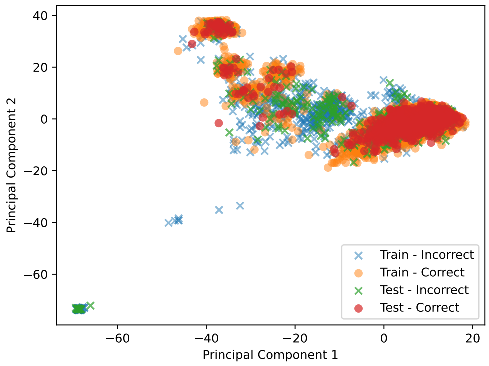


Figure 2: This visualization was created by performing 2D PCA on the hidden states of the last layer using the training data, and then projecting the test data onto the same dimensions. It reveals that there are certain regions in the middle where incorrect reasoning consistently appears.

3.2 Hidden States Probing

Objective. In this experiment, we train linear probes (logistic regression) and non-linear probes (SVM with RBF) on hidden layer representations to predict the correctness of the target model’s final answer for any given mathematical reasoning problem. Specifically, we want to study whether the hidden representations can already indicate the correctness of the final answer generated by the model as shown in Figure 2. High accuracy/F1-scores of the probes for specific layers would indicate that they could be where potential incorrectness gets added, resulting in incorrect responses from the models.

Experimental details. We select logistic regression classifiers as our probes for individual layer representations. Our training dataset consists of the layer representations for the final answer token, obtained when the target model generates responses for the elements in the training set, and labels which check the correctness of the final answers. We train separate probes for each layer and test their performance on the test set representations. Specifically, we query each sample from the test set from the target model and obtain the layer representations, which are fed into the probes to predict the correctness of the final generated answer. The model’s answer is compared against the ground truth answer, and the probes are evaluated on whether they can accurately predict the correctness/incorrectness of the model’s final answer. We measure the performance of the probes using standard accuracy and

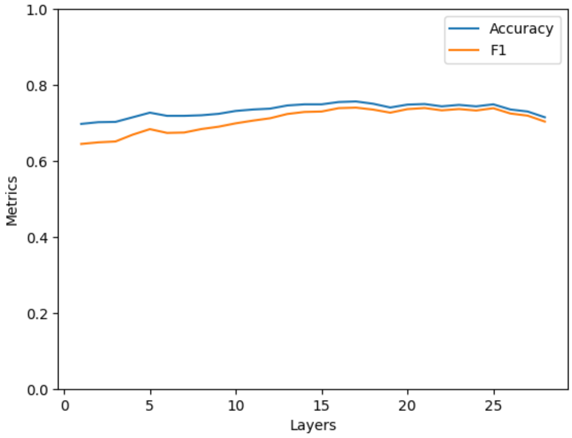


Figure 3: Accuracy and F1-scores of linear probes on hidden layers of Llama-3.2-3B-Instruct

Model	Accuracy (%)	F1 Score
LR	74.00	0.7331
SVM	70.66	0.6491

Table 1: Classification performance on the test data based on the last layer hidden state. Logistic Regression (LR) shows higher accuracy and F1 score compared to SVM.

F1-score metrics.

Results. Figure 3 shows the performance of the probes on the target model Llama-3.2-3B-Instruct for GSM-8K training and test sets with logistic regression. As shown in Figure 4, while the use of SVM allows for the visualization of the decision boundary, it demonstrates a lower F1 score, as indicated in Table 1. This supports the idea that the model's internal representation exists in a linear form, consistent with the linear representation hypothesis.

We see that the probes for different layers achieve high accuracy and F1-scores, hence indicating that all layers similarly contribute to the correctness of mathematical reasoning. We see slight increase in the performance of the probes for the middle layers. High performance indicates the importance of the corresponding layer's representations in determining the correctness of the final answer. Information on the importance of layers can provide insights into the workings of the target model, thus constituting *global explanations* for the model's behavior on mathematical reasoning. Moreover, this information can guide the downstream application of representation editing, wherein the representations of important layers are modified to improve the target model's performance on mathematical reasoning tasks.

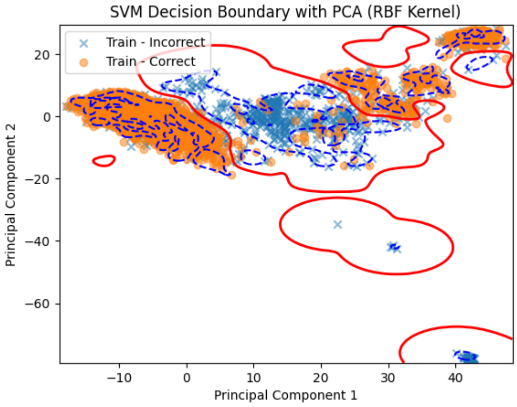


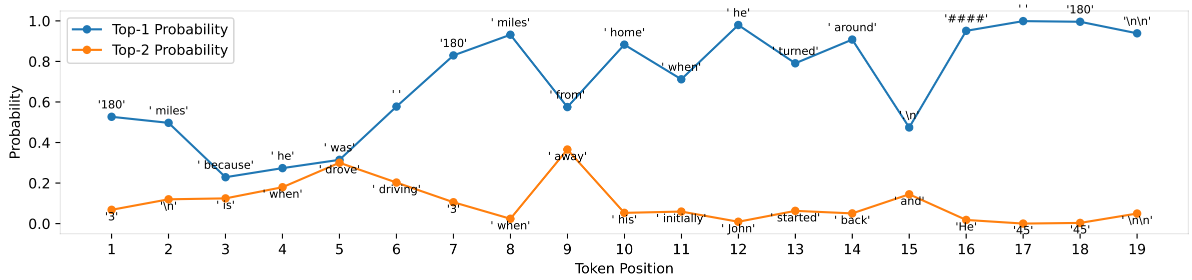
Figure 4: The decision boundary observed when applying SVM as a probe classifier to the normalized last hidden state of the training dataset. The red line represents the decision boundary where the SVM's decision function $f(\mathbf{x}) = 0$, while the blue dashed lines represent the margin boundaries where $f(\mathbf{x}) = \pm 1$. The region enclosed by the red line represents the "*incorrect area*," identifying where incorrect reasoning can be classified. In this experiment, the SVM achieved an accuracy of 70.66% and an F1 score of 0.6491.

3.3 Logits Probing

Objective. To take it a step further, the motivation for using logits to probe mathematical reasoning accuracy stems from two key ideas: (1) According to Li et al. (2023), the complexity of classifier matters, and a simple regressor may not be sufficient to uncover the representations within the model; and (2) a pre-trained linear layer with a softmax function decodes the last hidden layer into logits, which are designed to represent the probability of the model's prediction. This approach aims to address questions such as whether high-probability generations can be used to detect the model's incorrect answers.

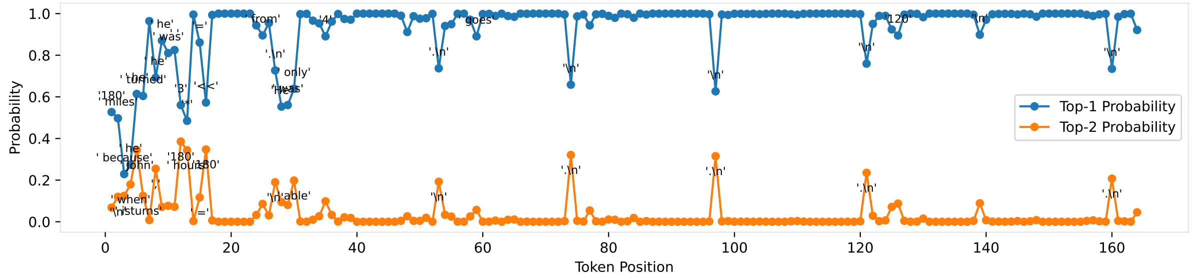
Experimental details. There are various ways to evaluate a model's certainty during generation using probabilities. For example, as illustrated in Fig. 5, we employed the following metrics using logits to differentiate between correct and incorrect reasoning and classify the correctness of the answer:

For example, as illustrated in Figure 6, we employed the following metrics to differentiate between correct and incorrect reasoning and classify the correctness of the answer. p_i denotes the array of generated token probabilities for the i-th ranked candidate:



Incorrect Answer (Top-k: 1, Greedy Search)

180 miles because he was 180 miles from home when he turned around. **#### 180**



Correct Answer (Top-k: 2, Sampling)

180 miles because when he turned around, he was 3*60=«3*60=180»180 miles from home. He was only able to drive 4-2=«4-2=2»2 hours in the first four hours. In half an hour he goes 30*.5=«30*.5=15»15 miles. He then drives another 2-.5=«2-.5=1.5»1.5 hours In that time he goes 80*1.5=«80*1.5=120»120 miles So he drove 120+15=«120+15=135»135 miles So he is 180-135=«180-135=45»45 miles away from home **#### 45**

Figure 5: The generation probabilities of incorrect answers generated with high probability (top) and correct answers achievable through sampling (bottom) when given a question like "John drives for 3 hours at a speed of 60 mph and then turns around because he realizes he forgot something very important at home. He tries to get home in 4 hours but spends the first 2 hours in standstill traffic. He spends the next half-hour driving at a speed of 30mph, before being able to drive the remaining time of the 4 hours going at 80 mph. How far is he from home at the end of those 4 hours?" This example shows that greedy search does not necessarily lead to better reasoning.

- Generated Length: One possible difference between correct and incorrect reasoning is the length of the response. A longer response may indicate a longer chain of thought (Wei et al., 2022), which could lead to better performance.
- $\mathbb{E}[p_1 - p_2]$: This represents the average difference between the top probability and the second-best candidate probability. If this value is low, the model faced attractive alternatives while generating the response.
- $\min(p_1 - p_2)$: This is the minimum difference between the probabilities of the top candidate and the second-best candidate across the sequence. A small value indicates moments of high uncertainty during generation.
- $\mathbb{E}[p_1]$: This is the average probability of the top candidate throughout the generation. Higher values suggest stronger overall confidence in the generated response.
- $\min(p_1)$: This is the minimum probability of the top candidate across the sequence, reflecting the least confident token in the response.

Corresponding to the probability-based metrics, we can also evaluate using logits, where z_i refers to the logits for the i-th ranked candidate:

- $\mathbb{E}[z_1 - z_2]$: The average difference between the logits of the top candidate and the second-best candidate.
- $\min(z_1 - z_2)$: The minimum difference between the logits of the top candidate and the

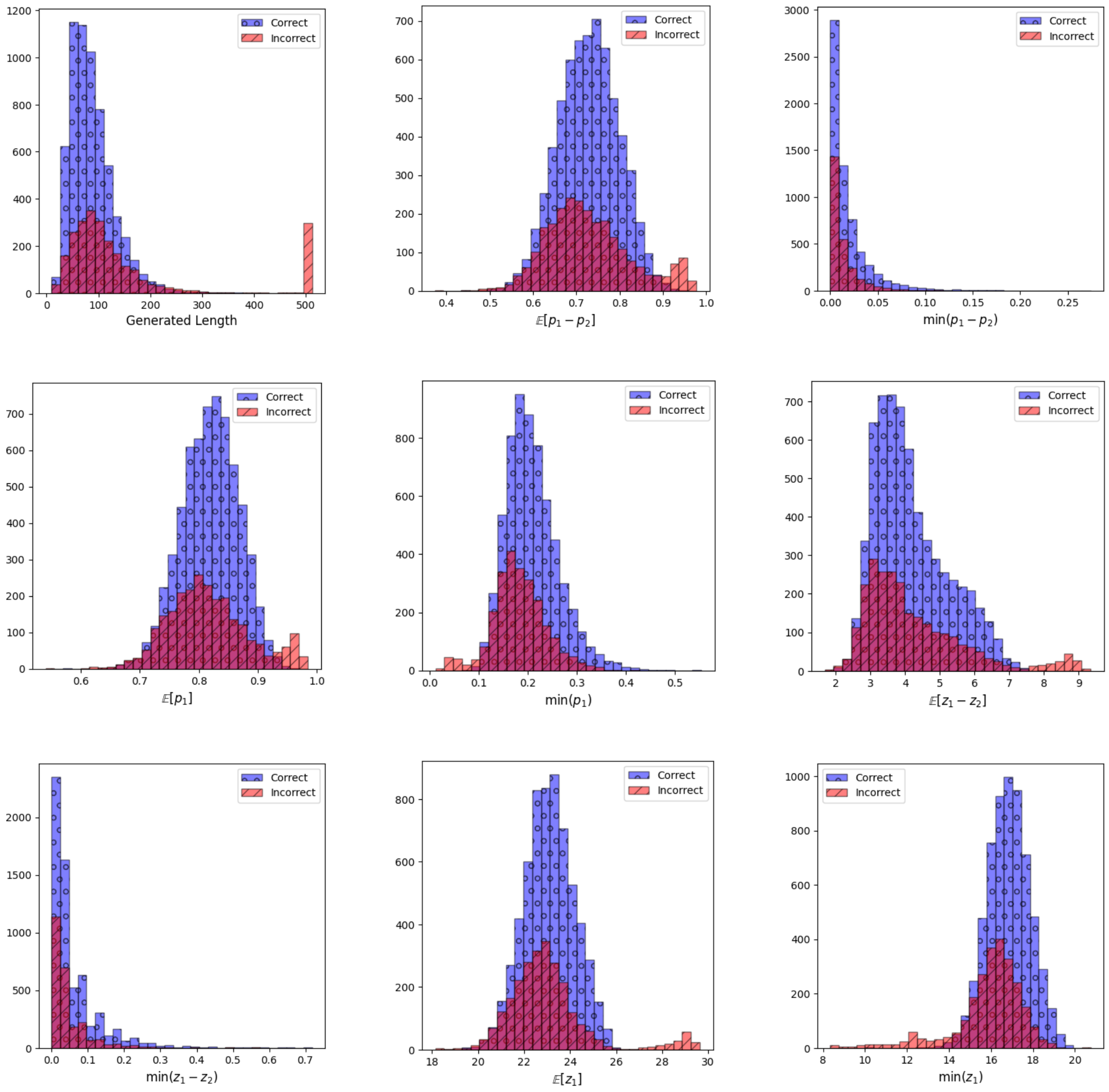


Figure 6: Histograms of metrics used for classification by probabilities (p_i) and logits (z_i). Cases where the model fails to perform proper reasoning and endlessly repeats tokens up to the maximum length of 512 are clearly distinguishable in the graph. This surprisingly led to experimental results where cases of reasoning with high confidence, also indicated by a large difference from the second candidate, were classified as incorrect. However, for more general cases, there is no apparent pattern that allows for reliable classification of incorrect reasoning.

second-best candidate.

- $\mathbb{E}[z_1]$: The average logit value of the top candidate across the sequence.
- $\min(z_1)$: The minimum logit value of the top candidate across the sequence.

Results. As shown in Fig. 6, cases where the generated length becomes abnormally long due to repetitive answer generation errors are clearly distinguishable. However, probing the token decoding process of the generated answer does not reveal a clear method to generally identify incorrect answers. When all these metrics are used as features for logistic regression, the model achieves an accuracy of 72.18% and an F1-score of 0.6849, which is not as high as the results obtained using hidden states.

3.4 Representation Engineering

Research aimed at controlling a model's representation based on probing was proposed by Zou et al. (2023) under the term "representation engineering."

Objective. The objective of our experiments was to extend the principles of Representation Engineering (RepE) to the domain of mathematical reasoning, aiming to investigate whether emergent representations of mathematical concepts and reasoning processes can be effectively located, monitored, and controlled within neural network models. Specifically, we sought to identify latent structures corresponding to mathematical operations, logical consistency, and problem-solving strategies, and to assess the feasibility of using these representations to improve model accuracy and interpretability in solving mathematical problems. By exploring these representations, our goal was to advance the transparency and controllability of large language models in handling tasks that require rigorous logical reasoning, thereby bridging gaps in current methodologies and contributing to the broader applicability of RepE in domains where precision and interpretability are critical.

Methodology. To extend Representation Engineering (RepE) to mathematical reasoning, we used the Mistral 7B v0.1 model and treated reasoning as a function. First, we built a dataset using GPT-4 and the GSM8K train split, generating examples of valid and invalid mathematical reasoning. These examples were crafted by prompting GPT-4 to produce reasoning steps and answers that were either logically correct or intentionally flawed.

Using this dataset, we elicited neural activity from the model by applying the prompt template:

> Answer the following question with <type=valid/invalid> mathematical reasoning: <question>

Results. Our experiments confirmed that such control not only fails to improve the model's accuracy but actually worsens it. We anticipate this might be due to the longer length of reasoning steps as opposed to true/false statements in . Zou et al. (2023). In order to better capture a representation for a correct mathematical answer, we perform additional experiments following (Wu et al., 2024). They divide a question to three or less subquestions, and answer them sequentially. This allows us to get a representation by using one line answers for each subquestion. While this approach was theoretically stronger, the control from this method too didn't work quite as expected.

4 Conclusion

We conducted analyses in various aspects of what occurs internally during reasoning in LLMs on the GSM8k dataset. Our experimental results show that some instances of incorrect reasoning can be classified with an F1 score exceeding 0.7. While we achieved partial success in classifying incorrect reasoning responses, a particularly intriguing finding emerged: the main feature distinguishing incorrect reasoning was not that longer reasoning chains improved accuracy, but rather that they often failed to converge on the correct answer. However, we also reveal that achieving a clean and reliable classification of incorrect reasoning using only the model's internal information, or improving the model's reasoning performance through probing, remains a challenging task, observing the intriguing result that the existing Representation Engineering (RepE) regime cannot be applied to mathematical reasoning and, instead, only worsens the outcomes.

Limitations

Our study does not distinguish between errors caused by inherent difficulty in the mathematical reasoning tasks and those resulting from systematic flaws or limitations in the model's architecture. While certain incorrect responses may stem from genuinely complex or ambiguous problems, others may be due to the model's inability to effectively leverage its internal representations. This limitation highlights the challenge of separating "difficult but plausible errors" from outright reasoning failures and emphasizes the need for future work to develop more nuanced evaluation methods to address this distinction.

Contributions for Each Group Member

Our team works together with everyone taking responsibility for all tasks. In particular, specific areas of responsibility of our team are:

- **Sewoong Lee** led this team as the team leader and first author, taking primary responsibility for coordinating all tasks, experiments, and presentations, ensuring that deliverables were met on time. He conceptualized the overall research direction, drafted most of the code, and wrote the majority of the manuscript. Additionally, he ensured consistency across team members' contributions by defining shared goals, establishing consistent experimental settings, and integrating their work.

- **Isha Chaudhary** worked on conceptualizing probing for mathematical reasoning and designing new probing experiments. Specifically, she took responsibility for extending the hidden states probing experiment to all layers.

- **Vinay Purushotham** worked on finding/creating contrastive datasets and setting up experiments. Additionally, he was responsible for replicating the RepE model, along with Vishal.

- **Vishal Bharadwaj** worked on visualization approaches. Additionally, he was responsible for replicating the RepE model, along with Vinay.

- **Sai Krishna Rohith K** worked on evaluating different models like Llama, Gemma. Specifically, he reproduced the GSM8k experiment with Mystral models, demonstrating that Llama is more suitable for this study.

Acknowledgements

This paper was written while Sewoong Lee works as a research assitant under Professor Julia Hockenmaier, and Isha Chaudhary was receiving guidance from Professor Gagandeep Singh. Lastly, we would like to express our deepest gratitude to Professor Heng Ji and TA Chi Han for their invaluable advice and guidance on our project, which helped us refine our output through revisions and enhance it with a more diverse perspectives in analysis.

References

Omer Antverg, Eyal Ben-David, and Yonatan Belinkov. 2022. Idani: Inference-time domain adaptation via neuron-level interventions.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2019. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek Abdelzaher, and Heng Ji. 2024. Word embeddings are steers for language models.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg.

2023. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model.

Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier.

Ryan Riegel, Alexander Gray, Francois Luus, Naweed Khan, Ndivhuwo Makondo, Ismail Yunus Akhalwaya, Haifeng Qian, Ronald Fagin, Francisco Barahona, Udit Sharma, Shajith Ikbal, Hima Karanam, Sumit Neelam, Ankita Likhyani, and Santosh Srivastava. 2020. Logical neural networks.

Zhuofeng Wu, He Bai, Aonan Zhang, Jiatao Gu, VG Vinod Vydiswaran, Navdeep Jaitly, and Yizhe Zhang. 2024. Divide-or-conquer? which part should you distill your llm?

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.