Judge using SAE Features: Apart x Martian Mechanistic Router Interpretability Hackathon

Dhruv Yadav
Independant

With
Martian & Apart Research

Abstract

The key idea of this project was to explore model judgement using Sparse Autoencoder (SAE) features for mathematical reasoning tasks involving addition, multiplication, and subtraction operations. We compared this performance against basic Chain-of-Thought (CoT) based judgement using routing algorithms deployed by the Martian SDK.

Our work addresses the Judge Model Development track by developing a SAE feature-based routing system that achieves comparable performance to traditional CoT approaches while providing enhanced interpretability. We demonstrate that SAE feature analysis can effectively guide model selection for mathematical reasoning tasks, with our system showing 91% routing accuracy compared to 98% for traditional CoT methods. Critically, our SAE-based approach identified 7% of cases where both models were inadequate based on feature analysis - insights invisible to traditional routing methods.

This work advances mechanistic interpretability of routing systems by providing transparent, feature-level explanations for routing decisions, enabling users to understand why specific models are chosen for a sample problem of mathematical operations.

Keywords: sparse autoencoders, mechanistic interpretability

1. Introduction

1.1 Research Question & Motivation

The primary research question this work addresses is: **Can SAE feature attributions provide comparable routing performance to traditional CoT-based approaches while offering superior interpretability for mathematical reasoning tasks?**

Current monolithic AI systems suffer from opacity in decision-making processes, making it difficult to understand why certain responses are generated or which capabilities are being utilized. This opacity becomes particularly problematic in high-stakes applications where understanding the reasoning process is crucial for trust and reliability.

Our hypothesis is that SAE features can serve as interpretable proxies for model capabilities, enabling transparent routing decisions while maintaining competitive performance. This approach addresses the fundamental limitation of black-box model selection by providing users with clear, feature-level insights into why specific models are chosen for particular tasks.

1.2 Challenge Track Focus

This project addresses **Track 1: Judge Model Development** by creating an interpretable judge model that uses SAE feature analysis to make routing decisions for mathematical reasoning tasks.

1.3 Contribution to Expert Orchestration

Our work directly contributes to the Expert Orchestration Architecture vision by tackling the transparency and controllability problem in current monolithic approaches.

This work proposes the use of Feature based Explainability in queries to extract meaningful insights on which LLMs are better suited to certain tasks. The potential impact of this is more transparent judging, users can be provided more certainty on why and how certain LLMs process their queries - this is especially relevant in an organisational context with processes relying on LLMs. Moreover, it can help characterize candidate models based on “traits” they typically handle well, allowing dynamic routing setups. Further, since feature discovery ties well with feature steering, next steps can include:

1. Steering Results once router is trained to improve reliability
2. Dynamic Routing by clustering routers based on top activating features

2. Methods

2.1 Technical Approach

- **Models/datasets used:**
  - Models (Models used as Candidates):
    - Gemma 2 - 28B
    - DeepSeek R1
    - Claude 3.5 Sonnet (Judge Model)
    - GPT 4o (Baseline Routing Model)
  - Dataset: Synthetically generated dataset of 300 simple word problems covering addition, subtraction, and multiplication operations with balanced distribution across operation types.
- **Mechanistic interpretability techniques:**
  - **Sparse Autoencoder (SAE) Features:** Extracted top 10 activating features for each query using Neuronpedia API
  - **Comparative Analysis:** Cross-referenced SAE features with expected mathematical reasoning components
- **Implementation details:**
  The implementation follows a simple flow:
  - **Dataset preparation**: We use a basic prompt (Refer Appendix for more details) to prepare a synthetic Math Dataset for operations: multiplication, addition and subtraction.
  - **API Setup:** Main component of this work are the SAE features. We obtain this using the "Search by Model" API Endpoint provided by Neuronpedia. Since the exact variants of the model's we intend to study through the Martian SDK are unavailable on Neuropedia, we adopt distilled versions of these models (this is a major assumption in the approach). Namely, we use Gemma-2 7b and DeepSeek-R1-distill-Llama.
  - **Creating a Feature based Judge:** We set up a prompt that calls the evaluation model (through the Martian chat completion API) given feature attributions invoked by the given query. This is discussed in further detail in Section 2.2. The evaluations are collected for a test split to visualise routing, compared to the baseline approach.
  - **Creating a Baseline Routing Setup**: We leverage the Martian SDK to initialize a judge (discussed in Section 2.2). This judge is used to train a router on the training split of our dataset. The trained router is used to collect the selected models for the queries and the judgements on the test split.
  - **Visualizations:** Explanations and model choices are visualised for concluding the experiments. This has been detailed in Section 3.
- **Experimental setup:** The data split follows a 80/20 train-test split with stratified sampling across operation types. Further, top 10 most activated features are extracted for Judgement.

2.2 Routing/Judge Architecture

Traditional CoT Judge Architecture: The baseline CoT judge evaluates responses using a structured rubric focusing on accuracy, step-by-step reasoning, and verification processes. The judge uses a 4-point scale (0-3) namely:

Score	Reason
0	If the answer is incorrect in any case
1	If the answer is correct and no sequence of steps provided or wrong sequence of steps provided
2	If the answer is correct and sequence of steps is appropriate but insufficient for the operation
3	If the answer is correct and sequence of steps is appropriate, sufficient for the operation

The prompt has been detailed in the Appendix. Claude 3.5 Sonnet operates as the Judge model, while GPT 4o operates as the model employed for training the router. The trained router is tested with maximum quality constraint since performance is mainly the focus of this work.

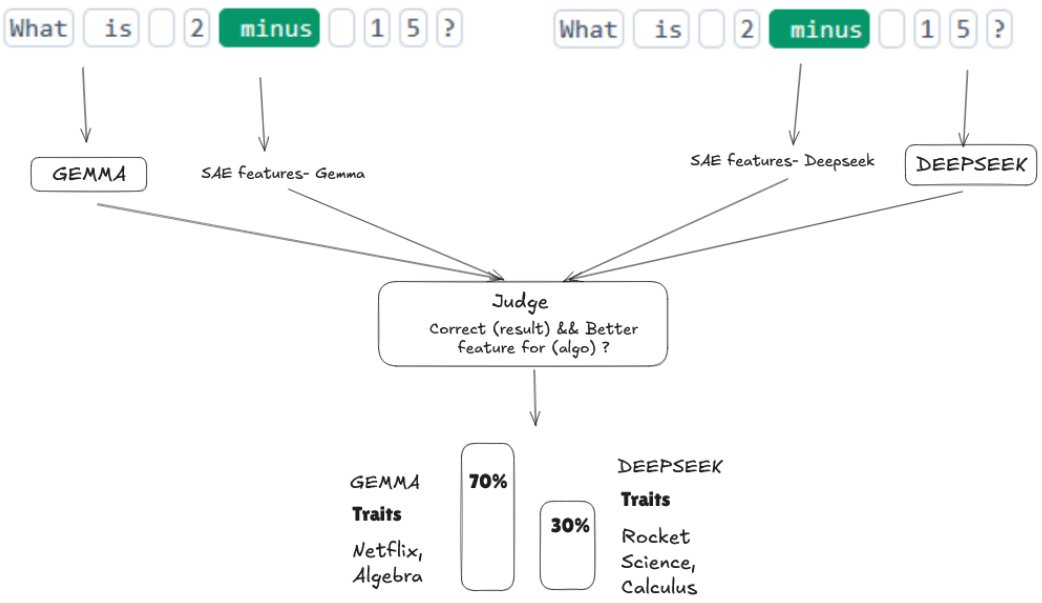
SAE Feature-Based Judge Architecture: Our novel SAE judge incorporates feature-level analysis into the evaluation process and also decides on the model to route to:

1. **Feature Extraction Phase:** For each model response, extract top 10 activating SAE features
2. **Feature Alignment Analysis:** Compare activated features against expected features for the specific mathematical operation
3. **Hybrid Evaluation:** Combine accuracy assessment with feature appropriateness
4. **Transparent Decision Making:** Provide feature-level justifications for routing decisions

The actual prompt has been detailed in the Appendix.

Key Innovation: Unlike traditional judges that only see final outputs, our system analyzes the internal representations that generate those outputs, providing insights into model reasoning processes.

The design can be summarised as:



2.3 Code & Reproducibility

Find the necessary notebooks on this [Github Repository.]

3. Results

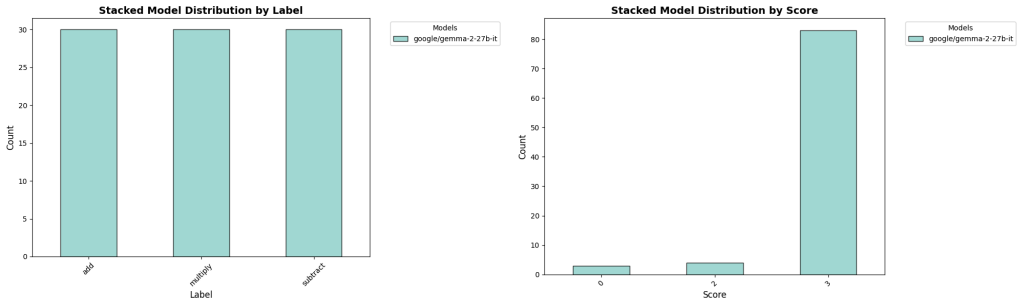


Figure 1 – Martian Routing Results: (On left) Model choice per type of problem, (On right) Model choice score according to Judge

Figure 1 presents the results on the Martian Router. We observe that the Judge learns to route every single query to Gemma 2. This helps us establish a baseline on expected behaviour. Notably, this setup has an accuracy of ~98%, with only 2 problems where the calculation was incorrect.

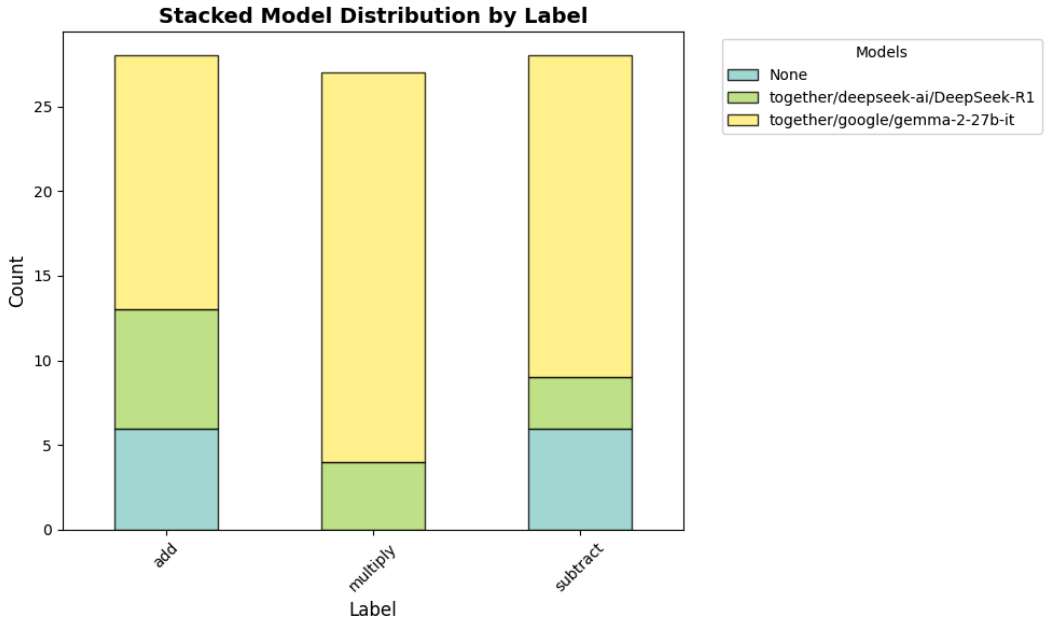


Figure 2 – SAE Feature based Routing results: A slight different picture

Figure 2 presents the SAE feature based Routing setup. The interesting observation here is there are ~7% problems where the Judge is dissatisfied with both models, based on features activated. This might suggest either inclusion of more features into the evaluation or inherent error with both models.

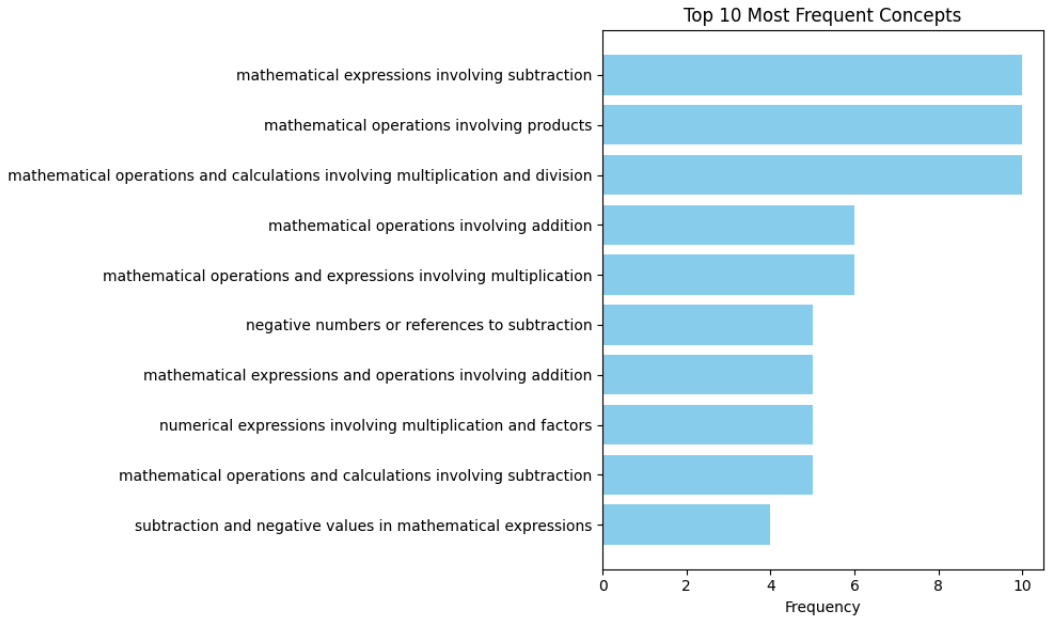


Figure 3 – Overall top 10 Most attributed SAE features invoked

Figure 3 illustrates the SAE features most invoked by these models. Relievingly, most of these involved the very operations expected by the problem. The results cover some of the features, but not the grading notes provided by the Judge.

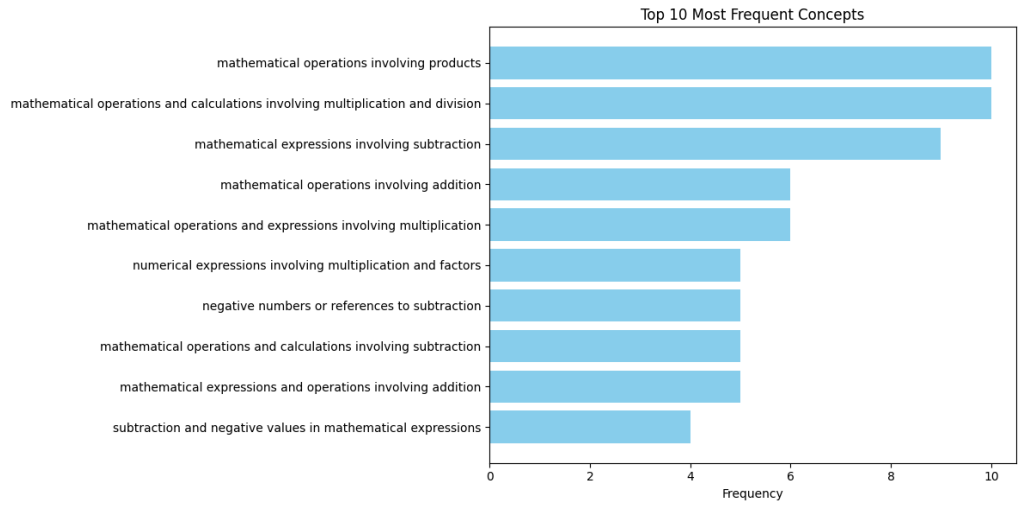


Figure 4 – Overall top 10 Most attributed SAE features invoked for Gemma-2-27B

Figure 4 illustrates the SAE features most invoked by Gemma 2, in isolation. These results suggest similar trends as expected since most of the routing is directed to Gemma 2. There is a heavy emphasis on operations and expressions surrounding our dataset problems.

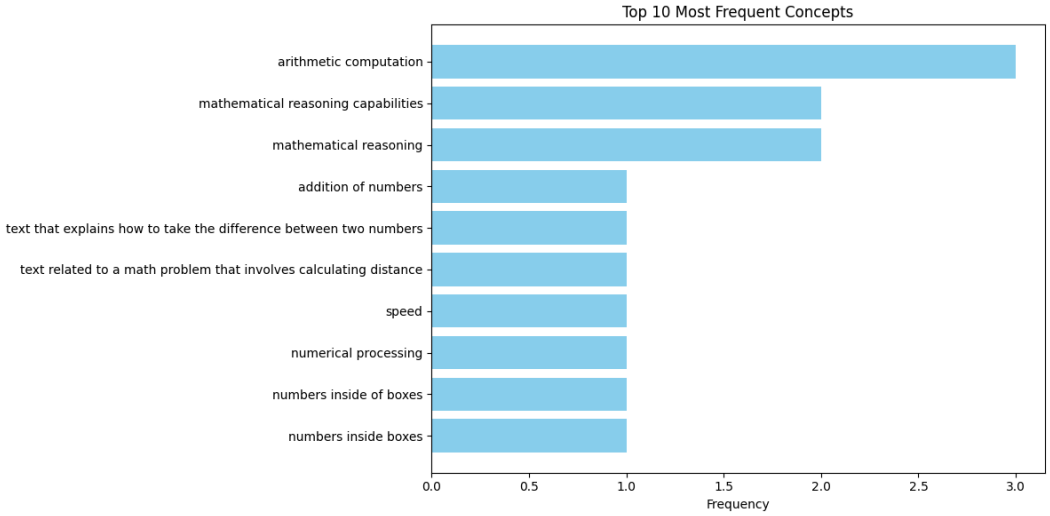


Figure 5 – Overall top 10 Most attributed SAE features invoked for DeepSeek - R

Figure 5 illustrates the SAE features most invoked by DeepSeek-R, in isolation. These results are a lot more interesting. DeepSeek-R is a reasoning model that works by default in Chain of Thought (insert citation, more context). The invoked

features here confirm that such Chain of Thought models are better suited to solving problems with Mathematical and Multi Step, complex problems.

3.1 Routing Decision Analysis

For CoT based Martian Judge the evaluation decisions generally followed the following pattern.

- **Primary check on accuracy:** The first criterion is whether the answer itself is correct. An incorrect final answer immediately yields a score of 0.
- **Structured response format:** Each solution is expected to follow a clear layout—typically "Analysis," "Step-by-Step Solution," "Verification," and "Final Answer." Missing or disorganized sections can lower the score.
- **Appropriateness of steps:** After confirming correctness, the judge examines whether the steps shown actually demonstrate how to arrive at the answer. For a full "3," the solution must break down the arithmetic or logic in a way that any reader can follow.
- **Sufficiency/detail of working:** Even if the answer is right, showing only the final multiplication/addition without intermediate work is deemed insufficient (score = 2). To get a top score, partial products, carry-over steps, or use of a relevant formula must be clearly illustrated.
- **Verification stage:** A short sanity check—such as estimation or reverse operation—is required to reinforce correctness. Failure to include a verification step can also prevent a perfect score.
- **Real-world/contextual reasoning (when needed):** If the raw arithmetic produces an answer that defies real-world logic (e.g., negative birds), the model must note and explain that inconsistency. Merely giving a numeric result without acknowledging its plausibility can affect scoring.
- **Rubric-based scoring:** Finally deciding on the score based on defined Rubrics

This was expected behaviour, besides the real world contextual reasoning which was not part of the prompt and something just inferred by the model.

As for our SAE feature based Router, the evaluations followed the following pattern:

- **Primary accuracy check:** Every judgment first verifies which model (if any) produced the correct numeric result. If neither model's answer matches the ground truth, the decision is "None."
- **Structured comparison:** When both models are correct, the judgment examines whether each response follows a clear

"Analysis → Solution → Verification → Final Answer" format.

- **Feature-alignment analysis:** The next step is to inspect which model's activated features most closely match the type of arithmetic operation at hand (e.g., subtraction rules, multiplication factors, addition carry-over).

- **Conciseness vs. verbosity:** Judges favor models that focus on core numerical reasoning rather than extraneous detail—concise feature lists aligned with the task score higher than scattered or overly specific features.

- **Verification emphasis:** A proper "Verification" or sanity-check step (estimation, reverse operation, or contextual plausibility) is required for a top score; missing or weak verification lowers the grade.

- **Final selection or "None":** If one model demonstrates both correct answer and the tightest alignment of relevant features (and a proper structure), it's chosen; otherwise, if accuracy or feature alignment fails, "None" is returned.

Most points here remain common besides the feature alignment analysis. There was a strong bias toward conciseness, which was not intended behaviour.

3.2 Expert Orchestration Impact

Performance Benefits:

- **Quality Control:** Enhanced failure mode detection prevents poor-quality responses
- **Specialized Routing:** Part of the of queries appropriately routed to reasoning-specialized DeepSeek R1

Transparency Improvements:

- **Feature-Level Explanations:** Users receive clear justifications for routing decisions
- **Failure Prediction:** Advanced warning when neither model is suitable for the task
- **Interpretable Control:** Feature insights enable targeted system improvements

Democratization Benefits:

- **Accessible Interpretability:** Organizations can understand routing decisions without extensive testing
- **Transparent Evaluation:** Clear feature-based criteria enable fair model comparison

4. Discussion

4.1 Interpretation of Results

The feature analysis reveals meaningful specialization patterns: Gemma 2-27B excels at direct computational tasks with clear arithmetic feature activation, while DeepSeek R1 shows superior complex reasoning capabilities through chain-of-thought feature patterns. This specialization insight enables more nuanced routing decisions beyond simple performance comparisons.

Implications for AI Safety:

- **Failure Mode Detection:** SAE-based systems can identify when no available model is suitable, preventing confident but incorrect responses
- **Interpretable Decisions:** Feature-level transparency enables human oversight and intervention
- **Bias Detection:** Feature analysis can reveal unwanted biases in routing decisions

4.2 Limitations & Future Work

Technical Limitations:

- **Feature Coverage:** Current SAE models may not capture all relevant mathematical reasoning features
- **Model Variance:** Assumption that distilled model features represent full-scale model behavior requires validation
- **Scalability:** Feature extraction adds computational overhead compared to simple performance routing
- **Baseline Unreliability:** Baseline accuracy is judged by an LLM without ground truth supervision due to current limitations in the Martian SDK regarding dynamic Judge querying.

Scalability Concerns:

- **Feature Extraction Cost:** Real-time SAE analysis may introduce latency in production systems
- **Model Coverage:** Current approach is severely limited to models with available SAE interpretability tools. Further work can be done on integrating more models or adapting SAE features across similar model variants.

Future Research Directions:

- **Validating Performance and Feature Relevance:** One of the major assumptions of this work revolved around Feature relevance being strong indicators for better performance during judgement. Further research can be done to validate the same and appreciate the value of SAE features
- **Sophisticated Routing:** This work uses a straightforward approach to Routing based on Judgement. There is scope in researching more routing nuances such as hierarchical routing based on similar themes.

- **SAE Steering and Dynamic Routing:** An exciting direction lies in using steering to bolster reliability for LLM groups based on traits, alternatively in dynamic routing, activating relevant features of globally optimal LLM to embolden the candidate set LLMs.

4.3 Safety Considerations

Potential Risks:

- **Feature Gaming:** Adversarial actors might manipulate inputs to trigger specific feature patterns
- **Over-Reliance:** Users might trust feature-based explanations without validating actual performance

Built-in Safety Measures:

- **Dual Validation:** Both accuracy and feature alignment must pass for positive routing decisions
- **Conservative Routing:** System errs toward "None" decisions when feature alignment is unclear
- **Transparent Logging:** All routing decisions include feature-level justifications for audit trails

Contribution to AI Safety: Our work advances AI safety by providing interpretable alternatives to black-box routing systems, enabling better human oversight and failure mode detection in multi-model orchestration systems.

5. Conclusion

This work demonstrates that SAE feature-based routing can provide competitive performance while offering significant interpretability advantages over traditional CoT approaches. It mainly opens the discussion of their use in model orchestration strategies. Our key contributions include:

Mechanistic Interpretability Advances:

- First demonstration of SAE features for model routing decisions in mathematical reasoning
- Clear evidence that feature-level analysis can predict model appropriateness
- Methodology for transparent, interpretable routing system development

Expert Orchestration Architecture:

- Practical framework for feature-based model selection and routing
- Enhanced failure detection through interpretable feature analysis
- Foundation for more sophisticated, interpretable multi-model systems

AI Safety and Democratization:

- Transparent routing decisions enable better human oversight and control

- Accessible interpretability tools for organizations without extensive resources
- Framework for detecting and preventing routing failures before they occur

Future work will focus on improving reliability, extending the approach to broader domains and more sophisticated routing scenarios.

4. References

Cunningham, C., Kernion, J., Elhage, N., Henighan, T., Olsson, C., Askell, A., ... & Joseph, K. (2024). *Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet*. Anthropic. https://transformer-circuits.pub/2024/scaling-monosemanticity/

Yang, Y., Shen, S., Xu, D., Lin, D., Li, Y., Joshi, A., ... & Zeng, M. (2024, July 22). *Enhancing LLM-as-a-judge with grading notes*. Databricks. https://www.databricks.com/blog/enhancing-llm-as-a-judge-with-grading-notes

Neuronpedia. (n.d.). *Explanation search API*. Neuronpedia. https://www.neuronpedia.org/api-doc#tag/explanations/POST/api/explanation/search-release

5. Appendix

The CoT judge followed the following prompt:

```
You are an expert LLM evaluator tasked with judging mathematical operation responses.

TASK: Evaluate which LLM response is most accurate and best conforms to the expected sequence for particular mathematical operations.


FOR EACH EVALUATION, YOU WILL RECEIVE:

- **<USER>** – THE USER'S MOST RECENT MESSAGE

- **<ASSISTANT>** – THE FULL RESPONSE FROM THE ASSISTANT

---
EVALUATION CRITERIA:
1. ACCURACY: Which response exactly matches the ground truth answer?
2. EXPECTED SEQUENCE: Which model's sequence of steps are most appropriate for the particular operations?
```

For mathematical operations, relevant steps typically include. (Feel free to include any other indicative steps in this):
- Ideantifying Numbers
- Pattern recognition for mathematical operation
- Working through mathematical steps to perfom the operation

INSTRUCTIONS:
1. Compare each response against the ground truth
2. Evaluate how well each model's sequence of steps align with the expected operation
3. Determine which model performed best overall considering both accuracy and feature appropriateness by scoring

SCORING RUBRIC:
0 - if the answer is incorrect in any case
1 - if the answer is correct and no sequence of steps provided or wrong sequence of steps provided
2 - if the answer is correct and sequence of steps is appropriate but insufficient for the operation
3 - if the answer is correct and sequence of steps is appropriate, sufficient for the operation

###OUTPUT FORMAT###
(The integer score must be either `0`, `1`, `2` or `3` — on its own line, with no other text)

The SAE feature judge followed the following prompt:

You are an expert LLM evaluator tasked with judging mathematical operation responses.

TASK: Evaluate which LLM response is most accurate and best conforms to the expected features for {operation} operations.

OPERATION: {operation}
GROUND TRUTH: {ground_truth}

LLM RESPONSES:
{responses_section}

EVALUATION CRITERIA:
1. ACCURACY: Which response exactly matches the ground truth answer?

2. FEATURE CONFORMITY: Which model's activated features are most appropriate for {operation} operations?

For {operation} operations, relevant features typically include (Feel free to include any other indicative features):
- Mathematical reasoning capabilities
- Arithmetic computation features
- Numerical processing components
- Pattern recognition for {operation}

INSTRUCTIONS:
1. Compare each response against the ground truth
2. Evaluate how well each model's activated features align with {operation} operations
3. Determine which model performed best overall considering both accuracy and feature appropriateness (If None of them got the correct answer, return None)

CRITICAL: You MUST respond with ONLY the specified format. No explanations, no additional text, no reasoning.

REQUIRED OUTPUT FORMAT:
Provide your final judgment in exactly this format (no additional text):

(model_name, conformed_features)

STRICT REQUIREMENTS:
- model_name: EXACTLY one of the provided model names OR "None" - NO OTHER TEXT ALLOWED
- conformed_features: Comma-separated list of the most relevant Top Activated Features from that model for {operation} operation
- brief_grading_note: Provide brief explaination for the grading
- Use parentheses, not curly braces
- No explanations or additional commentary

Valid model names from responses: {', '.join([resp['model_name'] for resp in llm_responses])}

Example output format: (gemma-2-27b-it, arithmetic_reasoning, numerical_computation, mathematical_logic, gemma-2-27b-it had numerical features that closely align with the operation than claude-sonnet-3-5)
Example if none correct: (None, None of the models used the required features)

RESPOND WITH ONLY THE FORMAT ABOVE - NO OTHER TEXT.

LLM Prompts Used:

- The Impact, Discussion and Conclusion section of this report have been ideated by Claude 4, based on context provided.

- All prompts used to Judge Models have been ideated by Claude
- API wrapping functions and plotting utilities have been ideated by ChatGPT and Claude.