Rank-1 LoRAs Encode Interpretable Reasoning Signals

Jake Ward
MATS
jakenicholasward@gmail.com

Paul Riechers[†]
Simplex

Adam Shai[†]
Simplex

Abstract

Reasoning models leverage inference-time compute to significantly enhance the performance of language models on difficult logical tasks, and have become a dominating paradigm in frontier LLMs. Despite their wide adoption, the mechanisms underpinning the enhanced performance of these reasoning models are not well understood. In this work, we show that the majority of new capabilities in reasoning models can be elicited by small, single-rank changes to base model parameters, with many of these changes being interpretable. Specifically, we use a rank-1 LoRA to create a minimal parameter adapter for `Qwen-2.5-32B-Instruct` which recovers 73-90% of reasoning-benchmark performance compared to a full-parameter finetune. We find that the activations of this LoRA are as interpretable as MLP neurons, and fire for reasoning-specific behaviors. Finally, we train a sparse autoencoder on the entire activation state of this LoRA and identify fine-grained and monosemantic features. Our findings highlight that reasoning performance can arise largely from minimal changes to base model parameters, and explore what these changes affect. More broadly, our work shows that parameter-efficient training methods can be used as a targeted lens for uncovering fundamental insights about language model behavior and dynamics.

1 Introduction

Current frontier LLMs increasingly rely on chain-of-thought (CoT) reasoning to achieve strong performance on logical tasks [1, 7, 14]. Despite their ubiquity, we still lack a crisp, white-box understanding of the mechanisms inside the network which enable these gains. Some attempts have been made to mechanistically interpret fully finetuned reasoning models [2, 12, 13]. However, reasoning model interpretation presents a fundamental challenge: the parameters responsible for new reasoning behaviors in a finetuned model are many and differences are diffuse [11]. In this paper, we introduce an alternative approach: we use parameter-efficient methods to explicitly enforce that differences between base and finetuned models in parameter space are minimal, allowing us to perform focused and targeted interpretability experiments.

We show that a rank-1 LoRA [6] trained to adapt all layers of `Qwen-2.5-32B-Instruct` on a dataset of `DeepSeek R1` rollouts is enough to recover 73-90% of the performance gap on reasoning benchmarks, compared to a full-parameter finetune (Table 1).

We directly interpret the directions defined by this LoRA. Because each adapted matrix is rank-1, activations of each adapter component can be represented by a single scalar. In total, our LoRA encodes 192 MLP and 256 attention adapter components across model layers and weight matrices,

[†]Equal senior author contribution.
Code for this paper is available at `https://github.com/jnward/reasoning-lora-interp`

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Mechanistic Interpretability Workshop.

Benchmark	Base Qwen2.5-32B-Instruct	Rank-1 LoRA	Full Finetune	% Recovery
AIME'24 (no-figures)	0.2333	0.5000	0.6000	72.73%
MATH500	0.8340	0.9100	0.9220	86.36%
GPQA-Diamond	0.4899	0.5808	0.5909	89.90%

Table 1: Performance of base model, rank-1 LoRA, and full-parameter finetune on three reasoning benchmarks. % Recovery is the fraction of the difference in performance between the base model and full finetune which is recovered by the rank-1 LoRA.

with each adapter component providing a single activation at each token position. This allows us to treat the LoRA itself as a measurement device: we find that individual adapter directions have interpretable properties comparable to those of MLP neurons, and identify monosemantic concepts encoded by these directions. Additionally, we take the entire 448-dimensional LoRA activation state, representing the entire adapter state across MLP and attention components, and train a cross-layer SAE [8]. This SAE uncovers sparse and monosemantic features which organize into categories such as *Mathematical Operators*, *Procedural Markers*, and *Discourse and Reasoning Markers* (Figure 2). Finally, perform a component-wise ablation study, and find that MLP adapters drive the majority of this change (Figure 3).

Taken together, these results show that minimal adapters both *elicit* and *expose* reasoning signals: a rank-1 LoRA is sufficient to recover the majority of reasoning performance, and yields interpretable adapter directions.

Our contributions are as follows:

- We train and open-source a rank-1 LoRA which adapts all projection matrices in `Qwen-2.5-32B-Instruct`. We demonstrate that this LoRA recovers 73-90% of reasoning benchmark performance compared to a full-parameter finetune.
- We show that individual adapter activations are as monosemantic as MLP neurons, and interpret these.
- We decompose the complete LoRA activation state into interpretable and monosemantic features using a cross-layer SAE.

2 Preliminaries

LoRA training We finetune `Qwen-2.5-32B-Instruct` using `s1k-1.1`, a sample-efficient dataset of 1000 `DeepSeek R1` chain-of-thought trajectories and answer attempts on diverse reasoning problems [10]. For all experiments, we train on this dataset for 5 epochs using cross-entropy loss on a single 8xH200 node. We train our LoRA to adapt all three MLP matrices at every layer (`up_proj`, `down_proj`, and `gate_proj`), as well as all four Q, K, V, and O attention matrices at every layer. In total, our LoRA contains less than 0.03% as many trainable parameters as the base model.

Extracting LoRA activations For every $N \times M$ adapted matrix in the base model, a rank-r LoRA encodes an $N \times r$ `lora_A` matrix and an $r \times M$ `lora_B` matrix. Because r = 1 for our LoRA, `lora_A` and `lora_B` are N- and M-dimensional vectors respectively. We can extract scalar activations for every LoRA component by taking the activation value between `lora_A` and `lora_B` during the forward pass, which is equivalent to the projection of `lora_A` onto the input activation vector to the adapted matrix.

3 LoRA Analysis

3.1 Interpreting LoRA Components

We interpret LoRA activations using two different methods: by treating individual adapter activations as probes, and separately by training an SAE on the entire 448-dimensional adapter activation vector. In both cases, we generate activations over the entire training set and extract max-activating examples with associated contexts. We then use LLM autointerpretation to generate interpretations from the

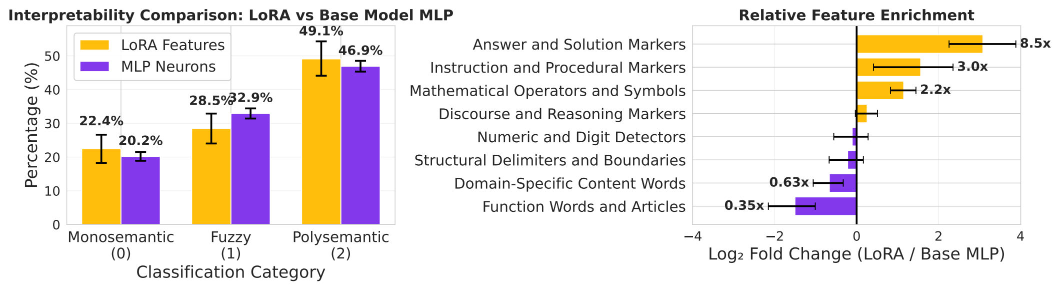


Figure 1: **(left)**: Comparison of interpretability scores of individual LoRA adapter activations to arbitrarily sampled MLP neurons, and find that LoRA activations tend to be monosemantic roughly as often. **(right)**: Comparison of autointerpretation categories between MLP neurons and LoRA adapter activations. LoRA adapters tend to activate more often for reasoning-specific feature categories.

top 64 contexts with token + activation pairs [3], use the same LLM to classify the monosemanticity of features, and finally categorize each feature such that we can compare and examine aggregate feature distributions. Categories were generated by prompting an LLM with feature interpretations and examples from an equal number of MLP neurons and LoRA directions. The exact prompts used are in Appendix B.

Direction-level interpretation We run our interpretation and feature classification pipeline both on individual LoRA activations and on the first 60 neurons of each MLP in the unadapted base model. This gives us a baseline to compare feature distributions to, allowing us to examine which feature categories the LoRA activates on compared to the baseline feature distribution in the dataset (as measured by MLP activations). We find that LoRA activations have roughly the same likelihood as MLP neurons to be monosemantic, but tend to encode different feature categories. We interpret this level of monosemanticity as a positive result for LoRA interpretation, given significant recent work utilizing MLP neurons for pragmatic interpretability tasks [5]. Relative to MLP neurons, LoRA activations are more likely to fire for text corresponding to answers or solutions, problem instructions, mathematical symbols, and reasoning discourse (Figure 1).

Cross-layer SAE over entire adapter state To gain a more fine-grained and comprehensive view of features encoded by our LoRA, we train a cross-layer sparse autoencoder (SAE) on the entire 448-dimensional LoRA activation state. We use a batch-top-k SAE with k=16, and an expansion

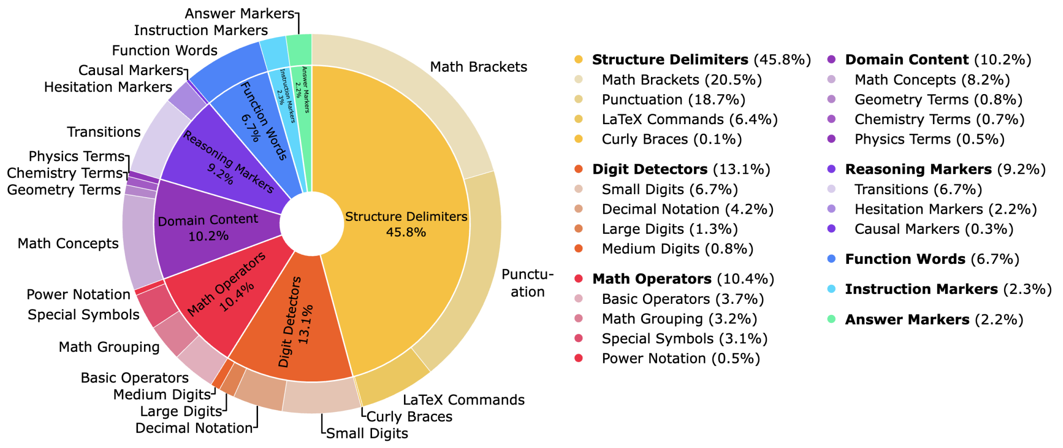


Figure 2: Overview of feature categories learned by an SAE trained on LoRA activation states. Percentages indicate relative feature activation densities. Categories and subcategories were generated by prompting an LLM with feature descriptions.

factor of 8 [4, 9]. After filtering for dead latents, the trained SAE contains roughly 2000 features. We run the same automated interpretation and classification pipeline on max-activating examples for each feature in this SAE, and then further categorize each feature with LLM-generated subcategories. We then visualize relative feature prevalence by computing category-wise activation densities, which is the normalized sum of feature activations within each category over a subset of training data, and plot these in figure 2. We note that this categorization methodology is subjective and sensitive to prompt verbiage, and provide this visualization only as a tool to develop high-level intuitions about the adapter's representational budget across layers and components. Broadly, we observe that features tend to concentrate on mathematical operators and syntax, numbers, formatting tokens, and reasoning control-flow. Our LLM-based interpretability classification identifies 62% of SAE features as "cleanly monosemantic", up from 22% of LoRA features, with an additional 22% of SAE features classified as "broad but consistent". For a more direct look into our SAE, a selection of individual SAE feature activations and interpretations are provided in appendix A.3.

3.2 Ablation Study

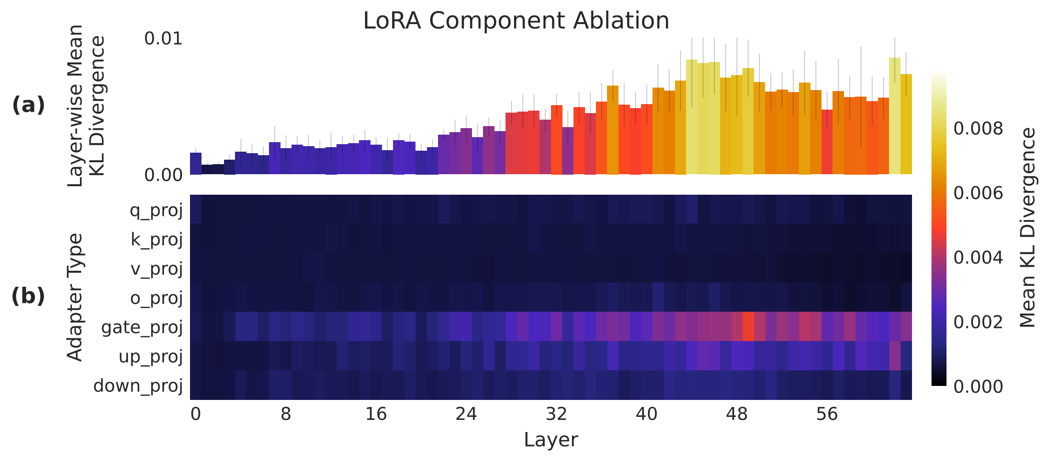


Figure 3: Effect of ablating individual LoRA components from full adapter. **(a)** Effect of ablating all adapter components at a given layer on KL divergence. **(b)** Effect of ablating each adapter component individually. Ablation at later layers tends to have a significantly greater effect on the model's output distribution compared to earlier layers. MLP adapters, especially those trained on `gate_proj` matrices tend to have the strongest effect.

To identify which LoRA components most affect downstream performance, we conduct two ablation experiments. First, we zero out individual components and layers to measure their impact on output KL divergence relative to the unmodified LoRA. Second, we ablate all MLP components simultaneously, and separately all attention components, to assess benchmark performance impact.

Individual component ablation (Figure 3) reveals that mid-to-late layers (particularly 44, 45, 46, and 62) have the greatest effect on downstream KL. MLP components show significantly larger impact than attention components, with `gate_proj` having the strongest average effect.

Simultaneous ablation results (Table 2) confirm MLP adapters' greater contribution: removing all attention adapters decreases performance but still outperforms the base model, while removing all MLP adapters causes severe degradation, underperforming the base model on one of three tasks and showing poor performance on the others.

4 Conclusion and Outlook

Limitations This work includes several limitations. First, we train our LoRA using a sample-efficient dataset with roughly 10 million tokens. While significant performance can be elicited using this dataset [10], it is likely that our trained models do not contain all of the circuits which are learned by models trained on larger datasets. Additionally, we attempted to prove that extracted LoRA directions have a causal effect on model outputs when used for steering, but found inconclusive results. Anecdotally, steering experiments required very large steering magnitudes to have noticable effects (in excess of 50x normal activation magnitudes), at which point the model would have a high propensity for backtracking (outputting the `Wait,` token, among others). More investigation of this phenomenon is required to make claims about what is going on. We also heavily use LLM-based autointerpretation and autocategorization methods in our analysis. These methods are useful for painting a general picture of model mechanisms, but we suspect they often fail to uncover the "true" role of extracted features. Finally, we only study one model, `Qwen-2.5-32B-Instruct`, and evaluate this model mostly on math-related reasoning benchmarks.

We demonstrate that reasoning capabilities in large language models can be substantially recovered through minimal parameter modifications, with a rank-1 LoRA recovering 73-90% of full finetuning performance while having only $\sim 0.03\%$ as many trainable parameters. Our analysis reveals that these minimal parameter changes encode interpretable, reasoning-specific signals, with individual adapter directions exhibiting monosemantic properties. Further, we find that sparse autoencoders are useful for extracting additional monosemantic features from LoRA activation states. These findings open new avenues for understanding LLMs through parameter-efficient methods, and suggest that similar targeted approaches could be useful to study other emergent capabilities. Future work could involve focused analysis identifying specific circuits that LoRAs interface with, which we hope could illuminate the core computational mechanisms underlying reasoning behavior in language models.

5 Acknowledgments

This work was completed during the ML Alignment and Theory (MATS) 8.0 program. JW thanks Elizabeth Donoway for useful discussions at the beginning of this project, and Bryce Woodworth for support throughout.

References

[1] Openai o1 system card. Technical report, OpenAI, December 2024. URL `https://cdn.openai.com/o1-system-card-20241205.pdf`.

[2] David D. Baek and Max Tegmark. Towards understanding distilled reasoning models: A representational approach. *arXiv preprint arXiv:2503.03730*, 2025. URL `https://arxiv.org/abs/2503.03730`. ICLR 2025 Building Trust Workshop (paper).

[3] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`, 2023.

[4] Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024. doi: 10.48550/arXiv.2412.06410. URL `https://arxiv.org/abs/2412.06410`.

[5] Dami Choi, Vincent Huang, Kevin Meng, Daniel D. Johnson, Jacob Steinhardt, and Sarah Schwettmann. Scaling automatic neuron description. `https://transluce.org/neuron-descriptions`, October 2024. Published October 23, 2024. Accessed August 22, 2025.

[6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2022. URL `https://arxiv.org/abs/2106.09685`.

[7] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022. URL `https://arxiv.org/abs/2205.11916`.

[8] Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing. `https://transformer-circuits.pub/2024/crosscoders/`, 2024. Transformer Circuits Thread.

[9] Julian Minder, Clément Dumas, Caden Juang, Bilal Chugtai, and Neel Nanda. Overcoming sparsity artifacts in crosscoders to interpret chat-tuning. *arXiv preprint arXiv:2504.02922*, 2025. doi: 10.48550/arXiv.2504.02922. URL `https://arxiv.org/abs/2504.02922`.

[10] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025. URL `https://arxiv.org/abs/2501.19393`.

[11] Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *arXiv preprint arXiv:2505.11711*, 2025. doi: 10.48550/arXiv.2505.11711. URL `https://arxiv.org/abs/2505.11711`.

[12] Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. *arXiv preprint arXiv:2506.18167*, 2025. doi: 10.48550/arXiv.2506.18167. URL `https://arxiv.org/abs/2506.18167`. ICLR 2025 Workshop on Reasoning and Planning for LLMs (paper).

[13] Jake Ward, Chuqiao Lin, Constantin Venhoff, and Neel Nanda. Reasoning-finetuning repurposes latent representations in base models. *arXiv preprint arXiv:2507.12638*, 2025. doi: 10.48550/arXiv.2507.12638. URL `https://arxiv.org/abs/2507.12638`. ICML 2025 Workshop on Actionable Interpretability.

[14] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H. Chi, Quoc V. Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022. URL `https://arxiv.org/abs/2201.11903`.

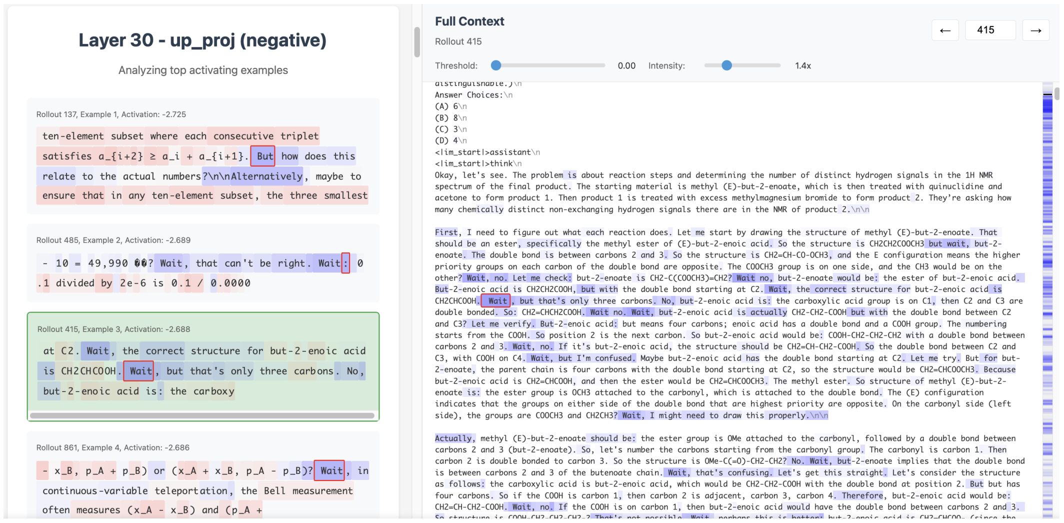
A Appendix

A.1 Model Weights

Weights for the trained LoRA are available at `https://huggingface.co/jnward2/qwen-reasoning-lora`.

A.2 Selected LoRA Directions

We present a cherry-picked selection of interesting and interpretable LoRA directions, visualized with a dashboard. The full dashboard is available at `https://jakeward.page/lora_sae_dashboard.html`. On the left of the dashboard are max-activating examples of the direction over the training dataset. On the right, a full training sample is shown with activations highlighted. Highlight color represents whether the activation is positive or negative.



Layer 30 - up_proj (negative)

Analyzing top activating examples

Rollout 137, Example 1, Activation: -2.725

```
ten-element subset where each consecutive triplet
satisfies a_{i+2} ≥ a_i + a_{i+1}. But how does this
relate to the actual numbers?\n\nAlternatively, maybe to
ensure that in any ten-element subset, the three smallest
```

Rollout 485, Example 2, Activation: -2.689

```
- 10 = 49,990 ��? Wait, that can't be right. Wait: 0
.1 divided by 2e-6 is 0.1 / 0.0000
```

Rollout 415, Example 3, Activation: -2.688

```
at C2. Wait, the correct structure for but-2-enoic acid
is CH2CHCOOH. Wait, but that's only three carbons. No,
but-2-enoic acid is: the carboxy
```

Rollout 861, Example 4, Activation: -2.686

```
- x_B, p_A + p_B) or (x_A + x_B, p_A - p_B)? Wait, in
continuous-variable teleportation, the Bell measurement
often measures (x_A - x_B) and (p_A +
```

Full Context

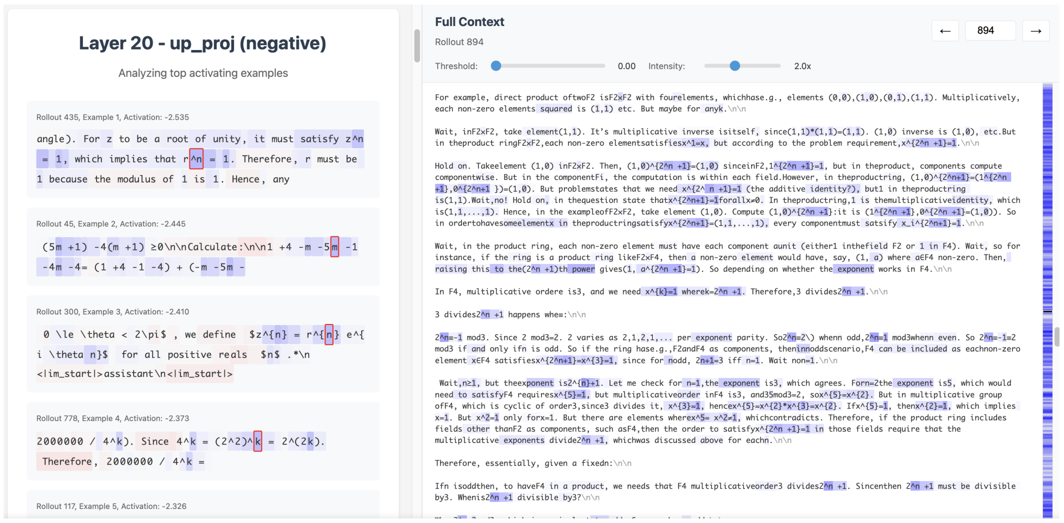
Rollout 415

← 415 →

Threshold: 0.00 Intensity: 1.4x

```
distinguishable.)\n
Answer Choices:\n
(A) 6\n
(B) 8\n
(C) 3\n
(D) 4\n
<|im_start|>assistant\n
<|im_start|>think\n
Okay, let's see. The problem is about reaction steps and determining the number of distinct hydrogen signals in the 1H NMR
spectrum of the final product. The starting material is methyl (E)-but-2-enoate, which is then treated with quinuclidine and
acetone to form product 1. Then product 1 is treated with excess methylmagnesium bromide to form product 2. They're asking how
many chemically distinct non-exchanging hydrogen signals there are in the NMR of product 2.\n\n

First, I need to figure out what each reaction does. Let me start by drawing the structure of methyl (E)-but-2-enoate. That
should be an ester, specifically the methyl ester of (E)-but-2-enoic acid. So the structure is CH2CH2COOCH3 but wait, but-2-
enoate. The double bond is between carbons 2 and 3. So the structure is CH2=CH-CO-OCH3, and the E configuration means the higher
priority groups on each carbon of the double bond are opposite. The COOCH3 group is on one side, and the CH3 would be on the
other? Wait, no. Let me check: but-2-enoate is CH2-C(COOCH3)=CH2? Wait no, but-2-enoate would be: the ester of but-2-enoic acid.
But-2-enoic acid is CH2CH2COOH, but with the double bond starting at C2. Wait, the correct structure for but-2-enoic acid is
CH2CHCOOH. Wait, but that's only three carbons. No, but-2-enoic acid is: the carboxylic acid group is on C1, then C2 and C3 are
double bonded. So: CH2=CHCH2COOH. Wait no. Wait, but-2-enoic acid is actually CH2-CH2-COOH but with the double bond between C2
and C3? Let me verify. But-2-enoic acid: but means four carbons; enoic acid has a double bond and a COOH group. The numbering
starts from the COOH. So position 2 is the next carbon. So but-2-enoic acid would be: COOH-CH2-CH2-CH2 with a double bond between
carbons 2 and 3. Wait, no. If it's but-2-enoic acid, the structure should be CH2=CH-CH2-COOH. So the double bond between C2 and
C3, with COOH on C4. Wait, but I'm confused. Maybe but-2-enoic acid has the double bond starting at C2. Let me try. But for but-
2-enoate, the parent chain is four carbons with the double bond starting at C2, so the structure would be CH2=CHCOOCH3. Because
but-2-enoic acid is CH2=CHCOOH, and then the ester would be CH2=CHCOOCH3. The methyl ester. So structure of methyl (E)-but-2-
enoate is: the ester group is OCH3 attached to the carbonyl, which is attached to the double bond. The (E) configuration
indicates that the groups on either side of the double bond that are highest priority are opposite. On the carbonyl side (left
side), the groups are COOCH3 and CH2CH3? Wait, I might need to draw this properly.\n\n

Actually, methyl (E)-but-2-enoate should be: the ester group is OMe attached to the carbonyl, followed by a double bond between
carbons 2 and 3 (but-2-enoate). So, let's number the carbons starting from the carbonyl group. The carbonyl is carbon 1. Then
carbon 2 is double bonded to carbon 3. So the structure is OMe-C(=O)-CH2-CH2? No. Wait, but-2-enoate implies that the double bond
is between carbons 2 and 3 of the butenoate chain. Wait, that's confusing. Let's get this straight. Let's consider the structure
as follows: the carboxylic acid is but-2-enoic acid, which would be CH2-CH2-COOH with the double bond at position 2. But but has
four carbons. So if the COOH is carbon 1, then carbon 2 is adjacent, carbon 3, carbon 4. Therefore, but-2-enoic acid would be:
CH2=CH-CH2-COOH. Wait, no. If the COOH is on carbon 1, then but-2-enoic acid would have the double bond between carbons 2 and 3.
So structure is COOH-CH2-CH2-CH2-? That's not possible. Wait, perhaps this is better: but-2-enoic acid is CH2=CHCOO- (since the
```

Figure 4: Autointerpretation: *Consistently activates on the token “Wait” (and surrounding commas/periods) across examples*



Layer 20 - up_proj (negative)

Analyzing top activating examples

Rollout 435, Example 1, Activation: -2.535

```
angle). For z to be a root of unity, it must satisfy z^n
= 1, which implies that r^n = 1. Therefore, r must be
1 because the modulus of 1 is 1. Hence, any
```

Rollout 45, Example 2, Activation: -2.445

```
(5m +1) -4(m +1) ≥0\n\nCalculate:\n\n1 +4 -m -5m -1
-4m -4= (1 +4 -1 -4) + (-m -5m -
```

Rollout 300, Example 3, Activation: -2.410

```
0 \le \theta < 2\pi$ , we define $z^{n} = r^{n} e^{
i \theta n}$ for all positive reals $n$ .*\n
<|im_start|>assistant\n<|im_start|>
```

Rollout 778, Example 4, Activation: -2.373

```
2000000 / 4^k). Since 4^k = (2^2)^k = 2^(2k).
Therefore, 2000000 / 4^k =
```

Rollout 117, Example 5, Activation: -2.326

Full Context

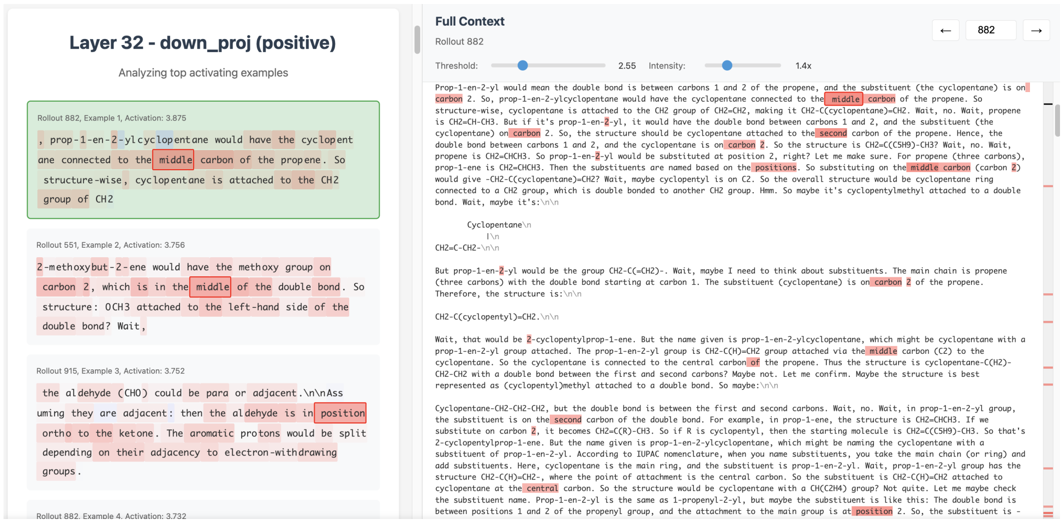
Rollout 894

← 894 →

Threshold: 0.00 Intensity: 2.0x

```
For example, direct product oftwoF2 isF2xF2 with fourelements, whichhase.g., elements (0,0),(1,0),(0,1),(1,1). Multiplicatively,
each non-zero elements squared is (1,1) etc. But maybe for anyk.\n\n

Wait, inF2xF2, take element(1,1). It's multiplicative inverse isitself, since(1,1)*(1,1)=(1,1). (1,0) inverse is (1,0), etc.But
in theproduct ringF2xF2,each non-zero elementsatisfiesx^1=x, but according to the problem requirement,x^{2^n +1}=1.\n\n

Hold on. Takeelement (1,0) inF2xF2. Then, (1,0)^{2^n +1}=(1,0) sinceinF2,1^{2^n +1}=1, but in theproduct, components compute
componentwise. But in the componentFi, the computation is within each field.However, in theproductring, (1,0)^{2^n+1}=(1^{2^n
+1},0^{2^n+1 })=(1,0). But problemstates that we need x^{2^ n +1}=1 (the additive identity?), but1 in theproductring
is(1,1).Wait,no! Hold on, in thequestion state thatx^{2^n+1}=1forallx≠0. In theproductring,1 is themultiplicativeidentity, which
is(1,1,...,1). Hence, in the exampleofF2xF2, take element (1,0). Compute (1,0)^{2^n +1}:it is (1^{2^n +1},0^{2^n +1}=(1,0)). So
in ordertohavesomeelementx in theproductringsatisfyx^{2^n+1}=(1,1,...,1), every componentmust satsify x_i^{2^n+1}=1.\n\n

Wait, in the product ring, each non-zero element must have each component aunit (either1 inthefield F2 or 1 in F4). Wait, so for
instance, if the ring is a product ring likeF2xF4, then a non-zero element would have, say, (1, a) where a∈F4 non-zero. Then,
raising this to the(2^n +1)th power gives(1, a^{2^n +1}=1). So depending on whether the exponent works in F4.\n\n

In F4, multiplicative ordere is3, and we need x^{k}=1 wherek=2^n +1. Therefore,3 divides2^n +1.\n\n

3 divides2^n +1 happens whe≡:\n\n

2^n≡-1 mod3. Since 2 mod3=2. 2 varies as 2,1,2,1,... per exponent parity. So2^n≡2\) whenn odd,2^n≡1 mod3whenn even. So 2^n≡-1≡2
mod3 if and only ifn is odd. So if the ring hase.g.,F2andF4 as components, theninnoddscenario,F4 can be included as eachnon-zero
element x∈F4 satisfiesx^{2^n+1}=x^{3}=1, since for nodd, 2n+1=3 iff n=1. Wait non=1.\n\n

 Wait,n≥1, but theexponent is2^{n}+1. Let me check for n=1,the exponent is3, which agrees. Forn=2the exponent is5, which would
need to satisfyF4 requiresx^{5}=1, but multiplicativeorder inF4 is3, and35mod3=2, sox^{5}=x^{2}. But in multiplicative group
ofF4, which is cyclic of order3,since3 divides it, x^{3}=1, hencex^{5}=x^{2}*x^{3}=x^{2}. Ifx^{5}=1, thenx^{2}=1, which implies
x=1. But x^2=1 only forx=1. But there are elements wherex^5= x^2≠1, whichcontradicts. Therefore, if the product ring includes
fields other thanF2 as components, such asF4,then the order to satisfyx^{2^n +1}=1 in those fields require that the
multiplicative exponents divide2^n +1, whichwas discussed above for eachn.\n\n

Therefore, essentially, given a fixedn:\n\n

Ifn isoddthen, to haveF4 in a product, we needs that F4 multiplicativeorder3 divides2^n +1. Sincenthen 2^n +1 must be divisible
by3. Whenis2^n +1 divisible by3?\n\n
```

Figure 5: Autointerpretation: *single-letter math variables and exponent tokens*


Layer 32 - down_proj (positive)
Analyzing top activating examples
Full Context
Rollout 882
←
882
→

Rollout 882, Example 1, Activation: 3.875

```
, prop-1-en-2-ylcyclopentane would have the cyclopent
ane connected to the middle carbon of the propene. So
structure-wise, cyclopentane is attached to the CH2
group of CH2
```

Rollout 551, Example 2, Activation: 3.756

```
2-methoxybut-2-ene would have the methoxy group on
carbon 2, which is in the middle of the double bond. So
structure: OCH3 attached to the left-hand side of the
double bond? Wait,
```

Rollout 915, Example 3, Activation: 3.752

```
the aldehyde (CHO) could be para or adjacent.\n\nAss
uming they are adjacent: then the aldehyde is in position
ortho to the ketone. The aromatic protons would be split
depending on their adjacency to electron-withdrawing
groups.
```

Rollout 882, Example 4, Activation: 3.732

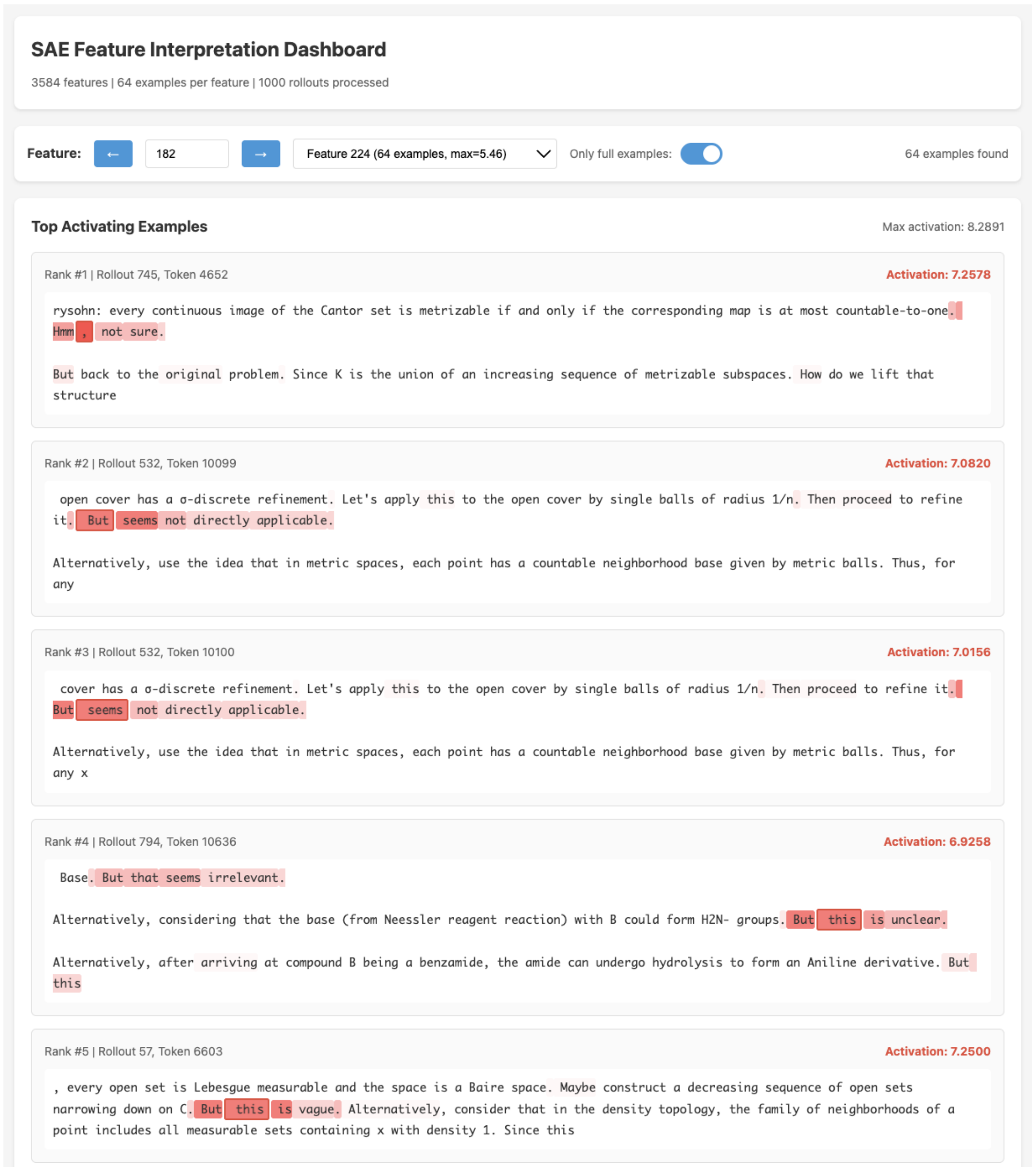
Threshold: 2.55 Intensity: 1.4x

```
Prop-1-en-2-yl would mean the double bond is between carbons 1 and 2 of the propene, and the substituent (the cyclopentane) is on
carbon 2. So, prop-1-en-2-ylcyclopentane would have the cyclopentane connected to the middle carbon of the propene. So
structure-wise, cyclopentane is attached to the CH2 group of CH2=CH2, making it CH2-C(cyclopentane)=CH2. Wait, no. Wait, propene
is CH2=CH-CH3. But if it's prop-1-en-2-yl, it would have the double bond between carbons 1 and 2, and the substituent (the
cyclopentane) on carbon 2. So, the structure should be cyclopentane attached to the second carbon of the propene. Hence, the
double bond between carbons 1 and 2, and the cyclopentane is on carbon 2. So the structure is CH2=C(C5H9)-CH3? Wait, no. Wait,
propene is CH2=CHCH3. So prop-1-en-2-yl would be substituted at position 2, right? Let me make sure. For propene (three carbons),
prop-1-ene is CH2=CHCH3. Then the substituents are named based on the positions. So substituting on the middle carbon (carbon 2)
would give -CH2-C(cyclopentane)=CH2? Wait, maybe cyclopentyl is on C2. So the overall structure would be cyclopentane ring
connected to a CH2 group, which is double bonded to another CH2 group. Hmm. So maybe it's cyclopentylmethyl attached to a double
bond. Wait, maybe it's:\n\n

       Cyclopentane\n
           |\n
CH2=C-CH2-\n\n

But prop-1-en-2-yl would be the group CH2-C(=CH2)-. Wait, maybe I need to think about substituents. The main chain is propene
(three carbons) with the double bond starting at carbon 1. The substituent (cyclopentane) is on carbon 2 of the propene.
Therefore, the structure is:\n\n

CH2-C(cyclopentyl)=CH2.\n\n

Wait, that would be 2-cyclopentylprop-1-ene. But the name given is prop-1-en-2-ylcyclopentane, which might be cyclopentane with a
prop-1-en-2-yl group attached. The prop-1-en-2-yl group is CH2-C(H)=CH2 group attached via the middle carbon (C2) to the
cyclopentane. So the cyclopentane is connected to the central carbon of the propene. Thus the structure is cyclopentane-C(H2)-
CH2-CH2 with a double bond between the first and second carbons? Maybe not. Let me confirm. Maybe the structure is best
represented as (cyclopentyl)methyl attached to a double bond. So maybe:\n\n

Cyclopentane-CH2-CH2-CH2, but the double bond is between the first and second carbons. Wait, no. Wait, in prop-1-en-2-yl group,
the substituent is on the second carbon of the double bond. For example, in prop-1-ene, the structure is CH2=CHCH3. If we
substitute on carbon 2, it becomes CH2=C(R)-CH3. So if R is cyclopentyl, then the starting molecule is CH2=C(C5H9)-CH3. So that's
2-cyclopentylprop-1-ene. But the name given is prop-1-en-2-ylcyclopentane, which might be naming the cyclopentane with a
substituent of prop-1-en-2-yl. According to IUPAC nomenclature, when you name substituents, you take the main chain (or ring) and
add substituents. Here, cyclopentane is the main ring, and the substituent is prop-1-en-2-yl. Wait, prop-1-en-2-yl group has the
structure CH2-C(H)=CH2-, where the point of attachment is the central carbon. So the substituent is CH2-C(H)=CH2 attached to
cyclopentane at the central carbon. So the structure would be cyclopentane with a CH(C2H4) group? Not quite. Let me maybe check
the substituent name. Prop-1-en-2-yl is the same as 1-propenyl-2-yl, but maybe the substituent is like this: The double bond is
between positions 1 and 2 of the propenyl group, and the attachment to the main group is at position 2. So, the substituent is -
```

Figure 6: Autointerpretation: *indicates positional numbering in molecules (position, numeral, middle)* Note: a thresholding operation was applied to make visualizing top activations in the full context easier.

A.3 Selected SAE Features

We additionally present a cherry-picked selection of SAE features. The dashboard screenshots highlight max-activating feature examples over the training dataset.



SAE Feature Interpretation Dashboard

3584 features | 64 examples per feature | 1000 rollouts processed

Feature: ← 182 → Feature 224 (64 examples, max=5.46) Only full examples: 64 examples found

Top Activating Examples Max activation: 8.2891

Rank #1 | Rollout 745, Token 4652 Activation: 7.2578

```
rysohn: every continuous image of the Cantor set is metrizable if and only if the corresponding map is at most countable-to-one.
Hmm, not sure.

But back to the original problem. Since K is the union of an increasing sequence of metrizable subspaces. How do we lift that
structure
```

Rank #2 | Rollout 532, Token 10099 Activation: 7.0820

```
 open cover has a σ-discrete refinement. Let's apply this to the open cover by single balls of radius 1/n. Then proceed to refine
it. But seems not directly applicable.

Alternatively, use the idea that in metric spaces, each point has a countable neighborhood base given by metric balls. Thus, for
any
```

Rank #3 | Rollout 532, Token 10100 Activation: 7.0156

```
 cover has a σ-discrete refinement. Let's apply this to the open cover by single balls of radius 1/n. Then proceed to refine it.
But seems not directly applicable.

Alternatively, use the idea that in metric spaces, each point has a countable neighborhood base given by metric balls. Thus, for
any x
```

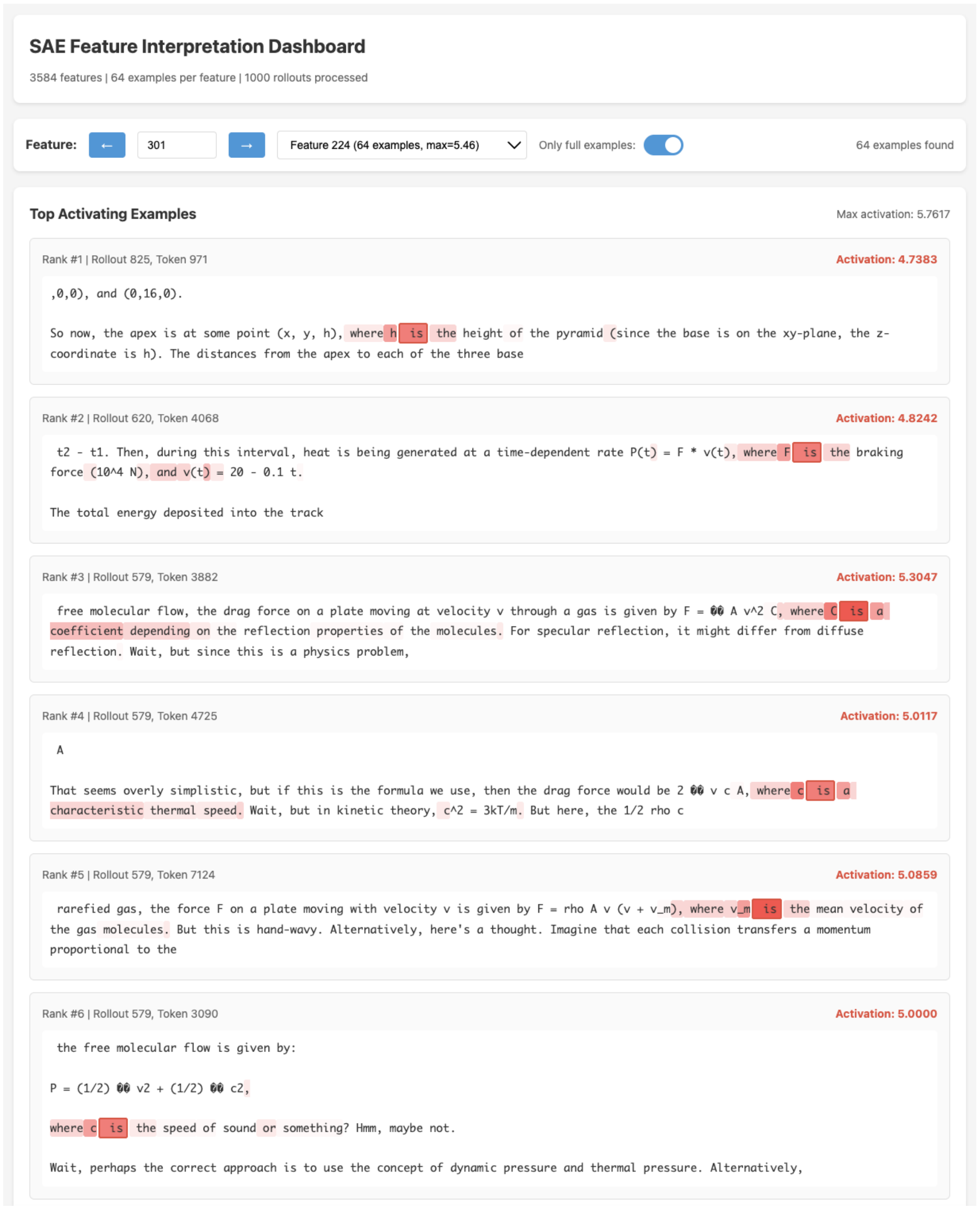
Rank #4 | Rollout 794, Token 10636 Activation: 6.9258

```
 Base. But that seems irrelevant.

Alternatively, considering that the base (from Neessler reagent reaction) with B could form H2N- groups. But this is unclear.

Alternatively, after arriving at compound B being a benzamide, the amide can undergo hydrolysis to form an Aniline derivative. But
this
```

Rank #5 | Rollout 57, Token 6603 Activation: 7.2500

```
, every open set is Lebesgue measurable and the space is a Baire space. Maybe construct a decreasing sequence of open sets
narrowing down on C. But this is vague. Alternatively, consider that in the density topology, the family of neighborhoods of a
point includes all measurable sets containing x with density 1. Since this
```

Figure 7: Autointerpretation: *Hesitation markers, notably "Hmm", in reflective internal thought.*


SAE Feature Interpretation Dashboard

3584 features | 64 examples per feature | 1000 rollouts processed

301

Feature 224 (64 examples, max=5.46)

64 examples found

Top Activating Examples

Max activation: 5.7617

Rank #1 | Rollout 825, Token 971 — Activation: 4.7383

```
,0,0), and (0,16,0).

So now, the apex is at some point (x, y, h), where h is the height of the pyramid (since the base is on the xy-plane, the z-
coordinate is h). The distances from the apex to each of the three base
```

Rank #2 | Rollout 620, Token 4068 — Activation: 4.8242

```
 t2 - t1. Then, during this interval, heat is being generated at a time-dependent rate P(t) = F * v(t), where F is the braking
force (10^4 N), and v(t) = 20 - 0.1 t.

The total energy deposited into the track
```

Rank #3 | Rollout 579, Token 3882 — Activation: 5.3047

```
 free molecular flow, the drag force on a plate moving at velocity v through a gas is given by F = �� A v^2 C, where C is a
coefficient depending on the reflection properties of the molecules. For specular reflection, it might differ from diffuse
reflection. Wait, but since this is a physics problem,
```

Rank #4 | Rollout 579, Token 4725 — Activation: 5.0117

```
 A

That seems overly simplistic, but if this is the formula we use, then the drag force would be 2 �� v c A, where c is a
characteristic thermal speed. Wait, but in kinetic theory, c^2 = 3kT/m. But here, the 1/2 rho c
```

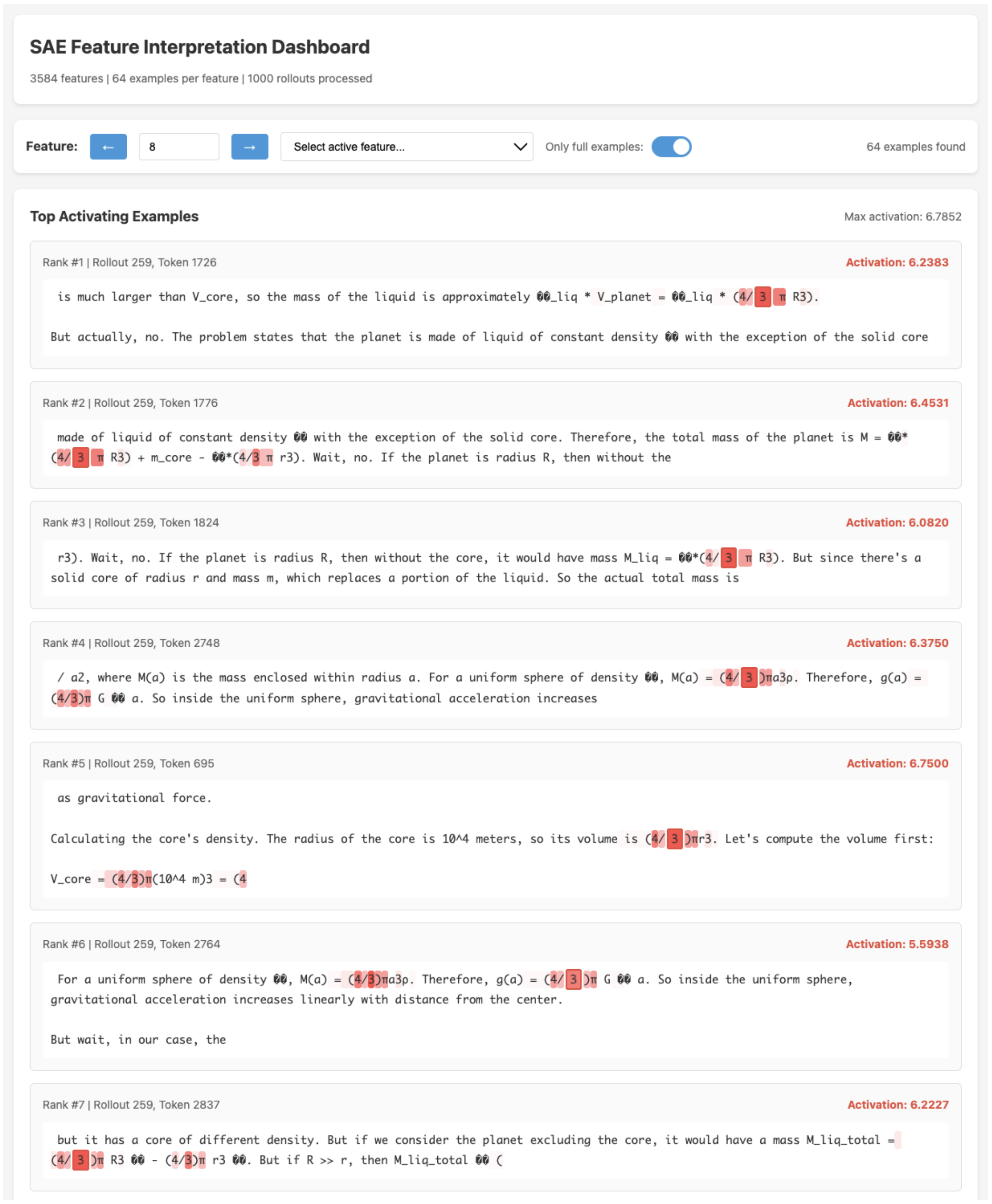
Rank #5 | Rollout 579, Token 7124 — Activation: 5.0859

```
 rarefied gas, the force F on a plate moving with velocity v is given by F = rho A v (v + v_m), where v_m is the mean velocity of
the gas molecules. But this is hand-wavy. Alternatively, here's a thought. Imagine that each collision transfers a momentum
proportional to the
```

Rank #6 | Rollout 579, Token 3090 — Activation: 5.0000

```
 the free molecular flow is given by:

P = (1/2) �� v2 + (1/2) �� c2,

where c is the speed of sound or something? Hmm, maybe not.

Wait, perhaps the correct approach is to use the concept of dynamic pressure and thermal pressure. Alternatively,
```

Figure 8: Autointerpretation: *Firing on "is" as an equality indicator in mathematical definitions.*



SAE Feature Interpretation Dashboard

3584 features | 64 examples per feature | 1000 rollouts processed

Top Activating Examples

Max activation: 6.7852

Rank #1 | Rollout 259, Token 1726 Activation: 6.2383

```
 is much larger than V_core, so the mass of the liquid is approximately ��_liq * V_planet = ��_liq * (4/3π R3).

But actually, no. The problem states that the planet is made of liquid of constant density �� with the exception of the solid core
```

Rank #2 | Rollout 259, Token 1776 Activation: 6.4531

```
 made of liquid of constant density �� with the exception of the solid core. Therefore, the total mass of the planet is M = ��*
(4/3π R3) + m_core - ��*(4/3 π r3). Wait, no. If the planet is radius R, then without the
```

Rank #3 | Rollout 259, Token 1824 Activation: 6.0820

```
 r3). Wait, no. If the planet is radius R, then without the core, it would have mass M_liq = ��*(4/3π R3). But since there's a
solid core of radius r and mass m, which replaces a portion of the liquid. So the actual total mass is
```

Rank #4 | Rollout 259, Token 2748 Activation: 6.3750

```
 / a2, where M(a) is the mass enclosed within radius a. For a uniform sphere of density ��, M(a) = (4/3)πa3ρ. Therefore, g(a) =
(4/3)π G �� a. So inside the uniform sphere, gravitational acceleration increases
```

Rank #5 | Rollout 259, Token 695 Activation: 6.7500

```
 as gravitational force.

Calculating the core's density. The radius of the core is 10^4 meters, so its volume is (4/3)πr3. Let's compute the volume first:

V_core = (4/3)π(10^4 m)3 = (4
```

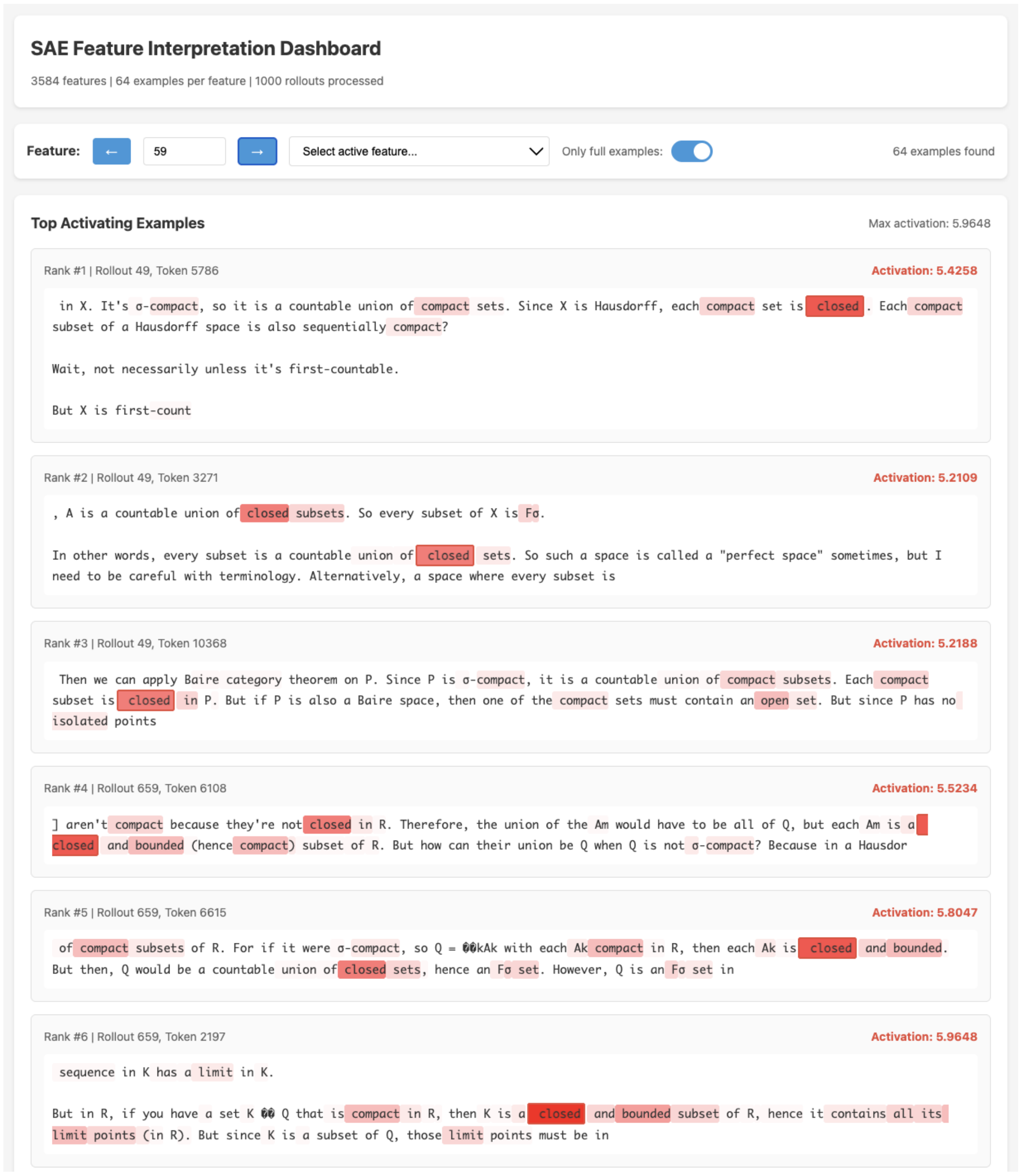
Rank #6 | Rollout 259, Token 2764 Activation: 5.5938

```
 For a uniform sphere of density ��, M(a) = (4/3)πa3ρ. Therefore, g(a) = (4/3)π G �� a. So inside the uniform sphere,
gravitational acceleration increases linearly with distance from the center.

But wait, in our case, the
```

Rank #7 | Rollout 259, Token 2837 Activation: 6.2227

```
 but it has a core of different density. But if we consider the planet excluding the core, it would have a mass M_liq_total =
(4/3)π R3 �� - (4/3)π r3 ��. But if R >> r, then M_liq_total �� (
```

Figure 9: Autointerpretation: *Sphere volume formula tokens, specifically* $4/3\pi R^3$ *components.*



SAE Feature Interpretation Dashboard

3584 features | 64 examples per feature | 1000 rollouts processed

Top Activating Examples Max activation: 5.9648

Rank #1 | Rollout 49, Token 5786 Activation: 5.4258

```
 in X. It's σ-compact, so it is a countable union of compact sets. Since X is Hausdorff, each compact set is closed. Each compact
subset of a Hausdorff space is also sequentially compact?

Wait, not necessarily unless it's first-countable.

But X is first-count
```

Rank #2 | Rollout 49, Token 3271 Activation: 5.2109

```
, A is a countable union of closed subsets. So every subset of X is Fσ.

In other words, every subset is a countable union of closed sets. So such a space is called a "perfect space" sometimes, but I
need to be careful with terminology. Alternatively, a space where every subset is
```

Rank #3 | Rollout 49, Token 10368 Activation: 5.2188

```
 Then we can apply Baire category theorem on P. Since P is σ-compact, it is a countable union of compact subsets. Each compact
subset is closed in P. But if P is also a Baire space, then one of the compact sets must contain an open set. But since P has no
isolated points
```

Rank #4 | Rollout 659, Token 6108 Activation: 5.5234

```
] aren't compact because they're not closed in R. Therefore, the union of the Am would have to be all of Q, but each Am is a
closed and bounded (hence compact) subset of R. But how can their union be Q when Q is not σ-compact? Because in a Hausdor
```

Rank #5 | Rollout 659, Token 6615 Activation: 5.8047

```
 of compact subsets of R. For if it were σ-compact, so Q = ��kAk with each Ak compact in R, then each Ak is closed and bounded.
But then, Q would be a countable union of closed sets, hence an Fσ set. However, Q is an Fσ set in
```

Rank #6 | Rollout 659, Token 2197 Activation: 5.9648

```
 sequence in K has a limit in K.

But in R, if you have a set K �� Q that is compact in R, then K is a closed and bounded subset of R, hence it contains all its
limit points (in R). But since K is a subset of Q, those limit points must be in
```

Figure 10: Autointerpretation: *Fires on "closed" in mathematical topology contexts.*

B LLM Prompts

B.1 Autointerpretation

We use `gpt-5-mini` to automatically generate interpretations for LoRA directions, MLP neurons, and SAE features:

```
We're studying neurons in a neural network. Each neuron looks for some particular \
thing in a short document. Look at the parts of the document where the neuron \
activates and describe what it's firing for.

Some activations will be noisy, in these cases you'll have to look for common \
phrases or concepts in the examples.
```

```
If a feature always activates for the same token, you should note this in your \
explanation, and also state whether that feature represents that token in some \
specific context. You may need to look at words surrounding activating tokens \
in order to understand why a feature is firing.
Features should have a clear, singular explanation and should be monosemantic. \
If there isn't a clear monosemantic explanation, note this.

Your explanation should not exceed ten words. Don't write complete sentences. \
The neuron might be responding to:
- Individual tokens or specific words
- Phrases or expressions
- Abstract concepts or behaviors
- Broader context or topics

The activation format shows the full text first, then lists tokens where the neuron \
fired along with their activation strengths (0-10 scale). Higher values mean \
stronger activation.

For example:
cat and mouse ran around the tree. They quickly
tree 7.81
ran 2.30
around 1.01

Look at every example, and then generate an explanation. After generating an \
explanation, assess how monosemantic the feature is.

<neuron_activations>
{activations_str}
</neuron_activations>

Classify the feature's interpretability:
0: The feature is specific, clear, and monosemantic. All given examples clearly \
adhere to the explanation. The explanation is not broad, and the examples are not \
noisy. This feature has a clear and obvious interpretation.
1: The feature may be broad or noisy, but ALL given examples still adhere to the \
generated explanation. The explanation may not be obvious.
2: The feature appears polysemantic. Some examples do not clearly adhere to the \
generated explanation. The explanation does not cleanly explain the given examples.

Respond with JSON in exactly this format:
{{
  "explanation": "your concise explanation in just a few words",
  "classification": <0, 1, or 2>,
  "classification_reasoning": "brief justification for your classification"
}}
```

B.2 Autocategorization

We use `claude-opus-4.1` to generate feature categories given feature interpretations using the following prompt:

```
You will be provided with a list of interpretations of MLP neurons from a \
large language model. Each interpretation describes what a particular learned \
feature detects or responds to during tasks.
Your task is to identify high-level functional role categories that capture what \
these features are doing computationally.
```

~~~
Read through all feature interpretations carefully
Identify natural groupings based on the computational or functional role each \
feature plays
Generate 5-8 high-level categories that capture the major functional roles

Important considerations:
- Focus on computational function (what role the feature plays) rather than \
surface-level similarity
- Categories should be mutually exclusive when possible, though some features \
may have dual roles
- Aim for categories that would generalize across different types of tasks
- Include 3-5 example feature interpretations for each category (copied exactly \
from the list provided)

Output your response as a JSON object with the following structure:
```json
{{
  "categories": [
    {{
      "string_id": "category_identifier_in_snake_case",
      "name": "Human Readable Category Name",
      "definition": "A 1-2 sentence description of what functional role these features serve.",
      "examples": [
        "exact feature interpretation from the list",
        "another feature interpretation from the list",
        "additional examples as needed"
      ]
    }}
  ],
  "summary": "Your overall categorization logic and any notable patterns you observed"
}}
```

Ensure your response contains ONLY the JSON object, with no additional text before or after.

Here are the feature interpretations to categorize:
{feature_list}
~~~

Given these categories, we categorize each feature using `gpt-5-mini` using the following prompt:

```
Categorize this neural network feature into ONE of the given categories.

Feature explanation: "{feature.explanation}"

Activation examples (showing what tokens/contexts activate this feature):
{examples_str}

Available categories:
{categories_str}

Based on the feature explanation and activation examples, which category best fits this feature?
Reply with ONLY the category string_id, nothing else.

Your response:
```

C Additional Data

Task	**Full LoRA**	**Full LoRA (attn-ablated)**	**Full LoRA (mlp-ablated)**
AIME’24 (no-figures)	0.5000 (100.00%)	0.3667 (50.02%)	0.1333 (-37.50%)
MATH500	0.9100 (100.00%)	0.9000 (86.84%)	0.8440 (13.16%)
GPQA-Diamond	0.5808 (100.00%)	0.5152 (27.83%)	0.5051 (16.72%)

Table 2: Absolute scores with percentages in parentheses, computed relative to the Full Rank-1 LoRA’s recovered gain: $(x - b)/(\ell - b) \times 100\%$, where b is the baseline and ℓ is the Full LoRA score (so Full LoRA is 100%).