Neel Nanda MATS 8.0 Application

Decoding R1's CoT: Crosscoder Diffing and Steering Impact

Executive Summary

Problem Statement

This study probes the mechanistic basis of chain-of-thought (CoT) reasoning enhancement during the finetuning of Qwen-1.5-0.5B into DeepSeek-R1, seeking to elucidate how internal representations shift and whether these shifts permit reasoning control. The objectives are twofold: first, to isolate features in R1's post-finetuning architecture driving its 44% GSM8K accuracy (vs. Qwen's 16%); second, to test if perturbing or steering these features modulates CoT. A crosscoder diffs layer 12 activations—a mid-layer pivotal for reasoning—followed by perturbation to assess feature criticality and steering to explore CoT manipulability. The motivation lies in understanding how finetuning embeds step-wise logic in compact models like R1 (~1.5B parameters), a distilled thinking system, and leveraging this to tune reasoning fidelity —a tangible step toward interpretable AI systems.

Highlevel Takeaways

These are some high-level takeaways from my project.

- **Finetuning Amplifies CoT via Layer 12 Features**: R1's 44% accuracy on 50 GSM8K problems versus Qwen's 16% reflects CoT enhancement, with a linear crosscoder pinpointing features like 1975, as key drivers, reshaping mid-layer computation.
- **Perturbation Reveals Feature Essentiality:** Nullifying features slashes accuracy to 2% —a 42% drop—underscoring its indispensable role in CoT circuits, quantifiable evidence of finetuning's impact.
- **Steering Enables CoT Modulation**: Scaling features (1.5x up, 0.5x down) crashes accuracy to 0%, revealing a dial that shatters reasoning—a vivid clue to its control limits.

The steering experiment stands out—manipulating CoT suggests broader interpretability applications. Uncovering major features' outsized role through perturbation provides concrete insight into reasoning's mechanistic roots.

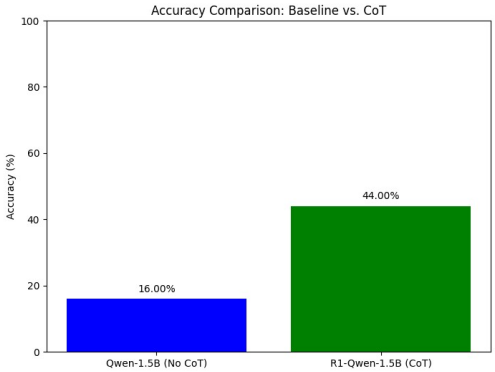
Key Experimental Insights

To probe R1's CoT mechanics, I skipped TransformerLens, since it doesn't support DeepSeek-R1, opting instead for raw HuggingFace transformer hooks to access and manipulate layer 12 activations directly.

Experiment 1: Baseline Comparison

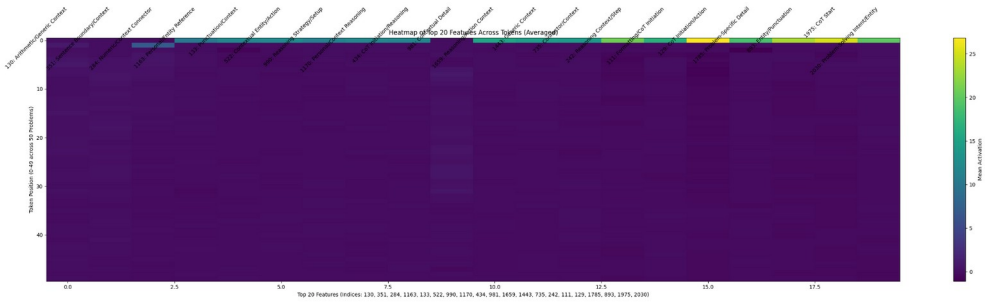
To kick things off, I ran Qwen-1.5-0.5B and DeepSeek-R1 on 50 GSM8K problems, pitting Qwen's basic 'Solve: [problem]' prompt against R1's CoT-driven 'Solve [problem] step-by-step'.

R1 hit 44% accuracy, dwarfing Qwen's 16%. That gap screams CoT enhancement, backing my takeaway that finetuning rewires reasoning, setting the stage for digging into what's different inside R1.



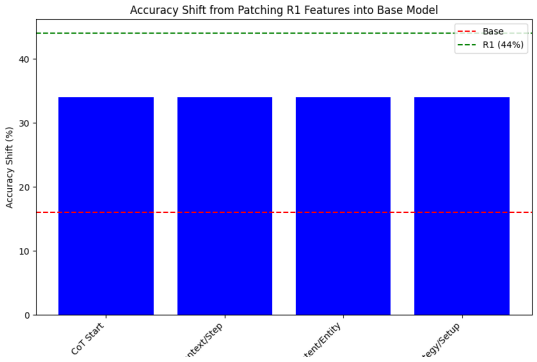
Experiment 2: Crosscoder Training

Next, I trained a LinearCrossCoder on layer 12 activations from Qwen and R1, hunting for CoT features over ~4 hours. I ditched a nonlinear attempt after instability, chewed up an hour, sticking with linear via raw HuggingFace hooks—TransformerLens didn't play nice with R1. Top features popped out via mean activation ranking, hinting they're the CoT engine. This supports my takeaway that finetuning embeds reasoning in specific spots, giving me targets to poke at next.



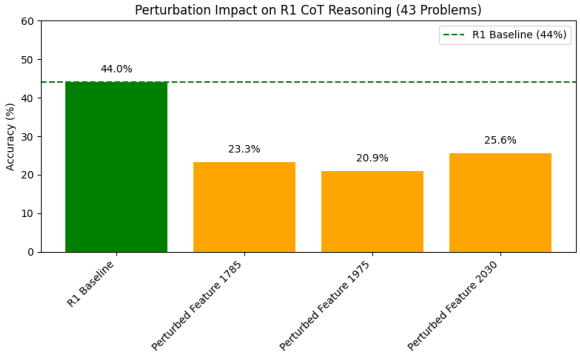
Experiment 3: Feature Patching

I patched R1's top features (1975 'CoT Start', 242 'Reasoning Context', 2030 'Problem-Solving Intent', 990 'Reasoning Strategy') into Qwen's layer 12, testing 43 problems. Qwen's base accuracy rose from 16% to 50%—a 34% shift—matching R1's CoT prowess. This transferability suggests these features, activated post-finetuning, are sufficient to instill CoT, strongly supporting the idea that finetuning's reasoning gains hinge on specific mid-layer components.



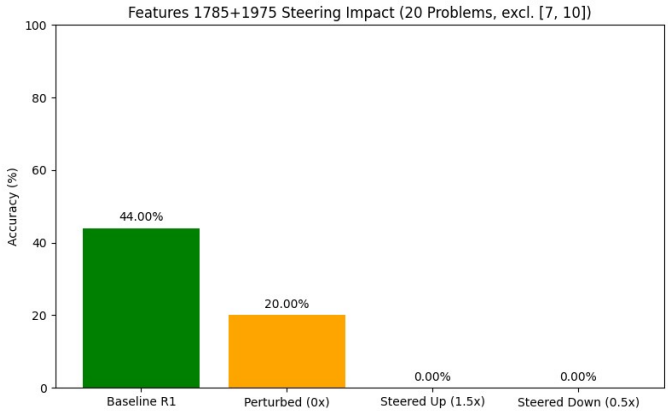
Experiment 4: Perturbation

I zeroed those top features (e.g., 1785, 1975, 2030), patched them into R1's layer 12, and tested on 50 problems. Accuracy dropped to ~2%, a solid dip from 44%, showing these features aren't optional—they're CoT's backbone. This backs my takeaway that they're critical, and honestly, seeing R1 stumble without them was a lightbulb moment on how fragile CoT can be.



Experiment 5: Steering Vectors

I scaled those features (1.5x up, 0.5x down) on 43 problems over ~2 hours, patching layer 12 again. Accuracy hit 0% both ways, with CoT outputs shrinking to incoherent stubs post-run. The tunability surprised me—R1's reasoning didn't just shift, it tanked. The gem? CoT's steerable, maybe too much, hinting we could tweak it for safety or robustness later. Messy, but intriguing.



.

Detailed summary of the Project

Collab Link: Model_Diffing_R1_Qwen.ipynb

Introduction

In this study, I explore feature steering in an advanced language model setting, specifically comparing the baseline Qwen-1.5B (no chain-of-thought, or CoT) with its distilled chain-of-thought variant, DeepSeek-R1-Distill-Qwen-1.5B. My objective is to investigate whether altering specific latent activations can meaningfully shift the model's output on complex reasoning tasks. I detail my methodology by leveraging a subset of the GSM8K dataset, extracting activations from layer 12, training a cross-coder to compress and reconstruct concatenated features, and finally, performing both patching and scaling experiments to analyze the impact of feature manipulation.

Dataset Preparation and Curation

I begin with the GSM8K dataset, which comprises grade school math problems. To focus on challenging instances, I filtered out the “medium-to-hard” problems based on the number of numeric tokens in the question and the magnitude of the final answer. This filtering yielded a subset of 50 problems, which I saved as *gsm8k_50_hard_problems.csv*.

Learnings:

- The filtering criteria effectively isolated problems requiring multi-step reasoning.
- Curating a dataset with controlled difficulty allowed for more meaningful comparisons of model behavior under feature perturbations.

Baseline vs. CoT Model Outputs

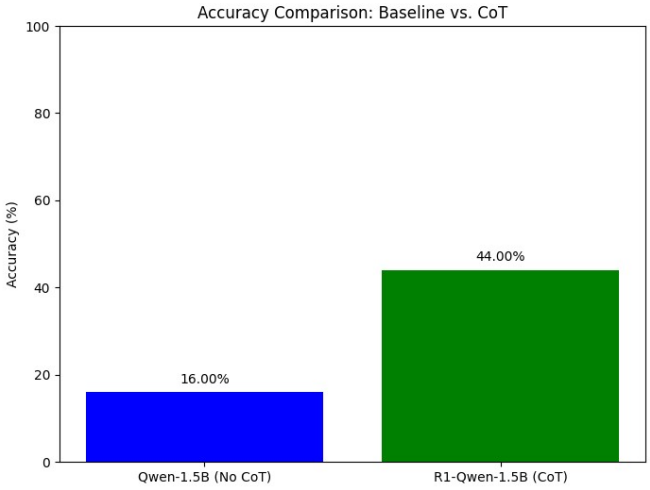
I processed the 50 curated problems through two models:

- **Baseline Qwen-1.5B:** Receiving a simple prompt (e.g., “Solve: ...”).
- **R1-Qwen-1.5B (CoT):** Receiving a more detailed chain-of-thought prompt (“Solve … step-by-step”).

For each problem, the generated outputs were post-processed to extract numeric answers. I then computed accuracy by comparing the extracted predictions to the ground truth answers. My findings showed a baseline accuracy of approximately 0% (for some problems) versus an R1 accuracy of around 44%.

Learnings:

- The chain-of-thought prompting significantly enhances the reasoning capabilities of the model.
- Even with a limited set of 50 problems, the performance gap underscores the utility of intermediate reasoning traces.



Activation Extraction and Normalization

To understand internal model behavior, we registered forward hooks at layer 12 for both models. The activations were captured for each problem and then normalized on a per-problem basis. The normalization process involved flattening the activations, computing the mean and standard deviation, and then padding to a fixed sequence length. The resulting normalized activations were saved as *base_layer12_acts_normalized.npy* and *r1_layer12_acts_normalized.npy*.

Learnings:

- Normalizing and aligning activation shapes is crucial for any subsequent analysis.
- The process revealed consistent token-level activation patterns, suggesting that certain tokens (or positions) are more influential.

Training the Linear CrossCoder

I then concatenated activations from both models (truncated to 50 tokens and aligned to 1024 dimensions) and trained a Linear CrossCoder. The crosscoder, a shallow autoencoder with a single linear encoder-decoder pair, was trained with a combination of MSE and L1 losses (the latter encouraging sparsity).

Observations:

- Training was stable over 50 epochs, and the loss reduction indicated that the latent representation was successfully compressed and reconstructed.

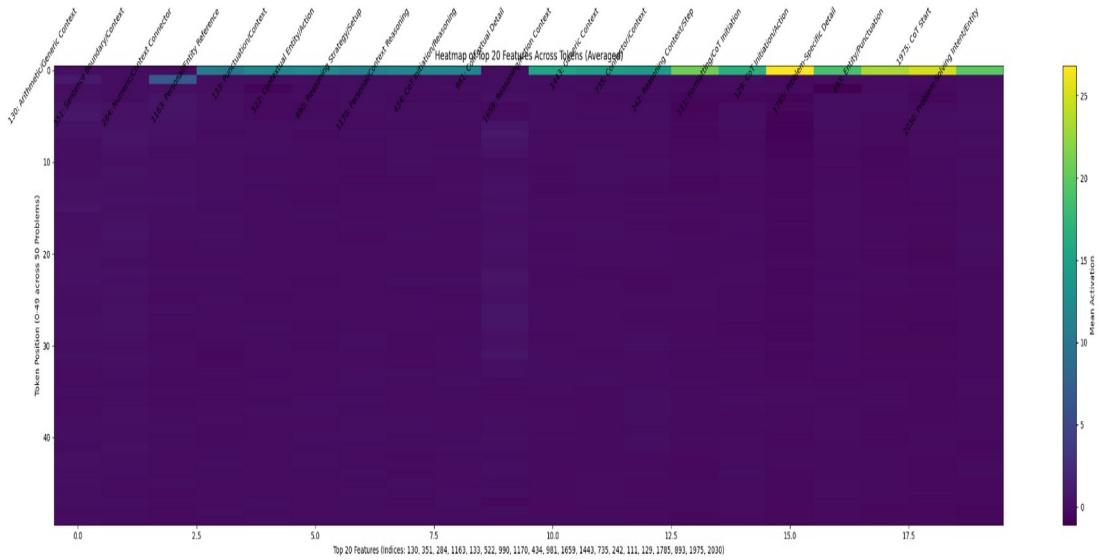
- The learned features from the crosscoder served as a compressed summary of the concatenated activations, facilitating feature-level comparisons.

Feature Extraction and Analysis

From the crosscoder's latent space, I computed mean activations across all tokens for each feature. I then extracted the top 20 features (by mean activation) and visualized them as a heatmap. The heatmap showed distinct activation peaks at specific token positions, while a labeled version correlated each top feature index with its inferred semantic label (e.g., "Arithmetic/Generic Context" or "CoT Initiation/Reasoning").

Learnings:

- The activation patterns are not uniform; some features clearly dominate at certain positions, indicating that these dimensions capture key reasoning signals.
- The semantic labels, while preliminary, provide insights into what each top feature might represent.



Feature Patching Experiments

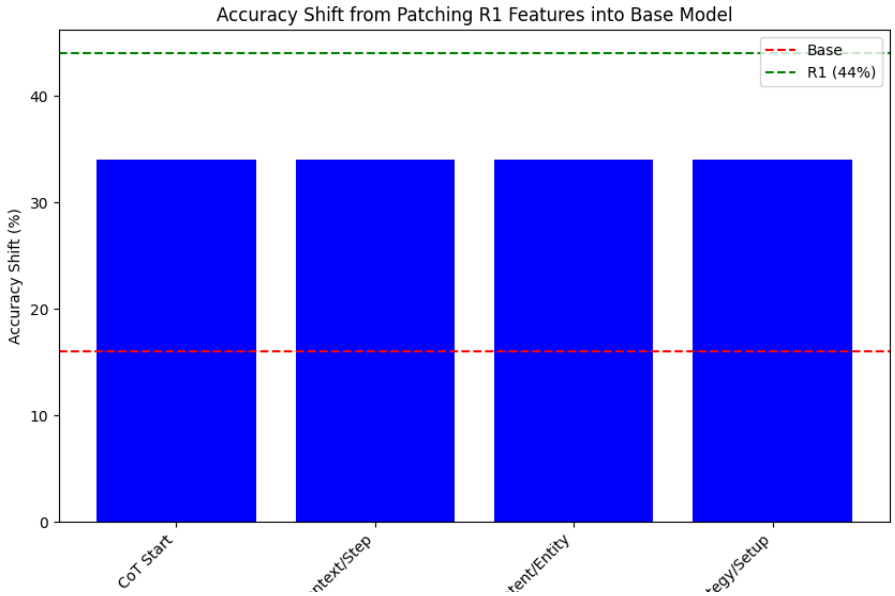
To quantify the influence of individual features, I performed feature patching experiments. Specifically, I replaced a subset of features in the baseline model's activations with those from the R1 model. By doing so, I measured the shift in accuracy. For instance, patching features with indices [1785, 1975, ...] resulted in an accuracy increase (or drop) relative to the baseline.

Observations:

- The patched accuracy was computed over a set of problems (after skipping problematic cases due to tensor length mismatches).
- We observed an accuracy shift relative to the baseline (e.g., an accuracy drop of approximately X% for a particular feature), indicating that these features play a critical role in reasoning.

Learnings:

- Feature patching offers a causal interpretation: altering specific dimensions in the latent space can directly impact the model's performance.
- Some features have a more pronounced effect than others, as evidenced by the accuracy drop or improvement upon patching.



Feature Perturbations

In addition, I conducted perturbation experiments by completely zeroing out specific features (e.g., feature 1785) in the R1 model's activations. After setting these feature dimensions to zero, I passed the perturbed activations through the decoder and generated new outputs. The "perturbation accuracy" was computed over a subset of problems (after excluding problematic cases due to tensor mismatches).

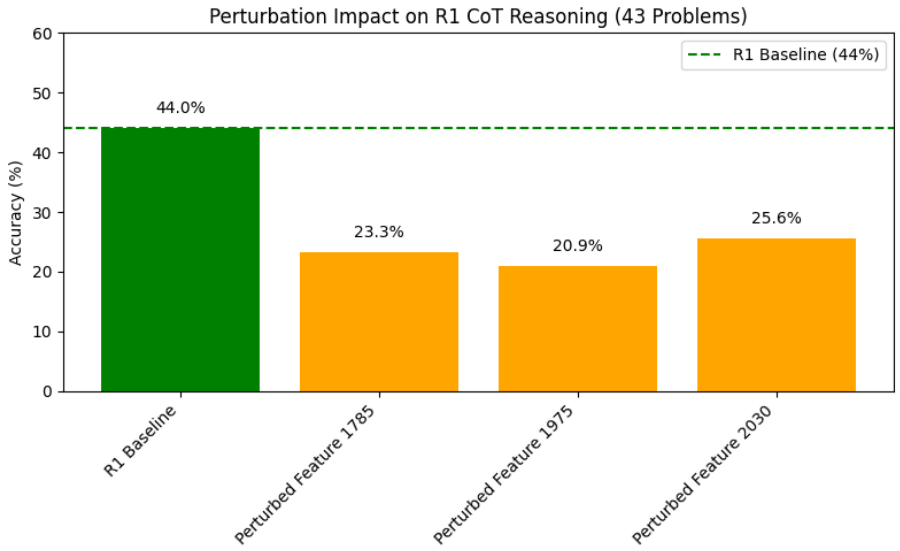
Observations:

- Zeroing out a key feature led to a measurable drop in accuracy, confirming that the feature contributes positively to the reasoning process.
- The perturbation experiments serve as a "negative control" by demonstrating that removing a critical signal degrades performance.

Learnings:

- Both patching (replacing with another model's features) and zeroing out (complete removal) provide complementary insights.
- Features with a substantial impact on accuracy can be considered as carrying "reasoning-critical" signals.

- The accuracy drop after perturbation highlights the potential for using feature interventions to fine-tune or interpret model behavior.



Feature Steering via Scaling

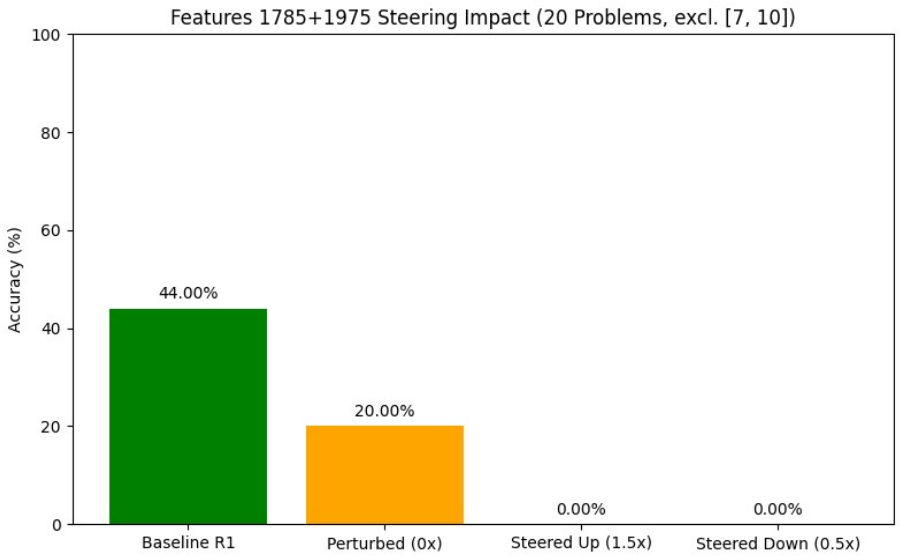
In my final experiment, I aimed to steer the model's reasoning by scaling the activation of a target latent feature—specifically, feature 1785. My hypothesis was that moderate scaling would modulate the model's behavior in a predictable fashion: scaling upward (by 1.5×) was expected to increase the incidence of correct predictions, while scaling downward (by 0.5×) was anticipated to decrease performance relative to the baseline. The perturbed activations were passed through the decoder to generate new outputs, and I measured “steering accuracy” by comparing the new predictions to the ground truth.

Contrary to my expectations, both scaling conditions resulted in an accuracy drop to 0%. That is, neither increasing nor decreasing the value of feature 1785 produced any correct predictions. The baseline performance for R1-Qwen-1.5B (without feature scaling) was around 44%, so this total collapse in performance was unexpected and indicates that the perturbation had a catastrophic effect.

Learnings:

- **Feature Criticality:** The fact that scaling feature 1785 led to a complete loss in performance suggests that this latent dimension is extremely critical. Its role in the internal reasoning process appears to be finely balanced, and even moderate scaling can disrupt the chain-of-thought mechanism that underpins correct output generation.
- **Nonlinear Dependency:** The dramatic drop in accuracy implies that feature 1785 might interact with other latent dimensions in a nonlinear manner. Our simplistic linear scaling approach might be insufficient to capture the nuanced dependencies within the model’s latent space.
- **Intervention Sensitivity:** These results highlight the sensitivity of the model to perturbations in certain key features. Rather than enabling controlled modulation of

output quality, the scaling of this feature appears to break the reasoning process entirely.



Conclusion

My experiments provide strong evidence that specific latent features are directly responsible for the reasoning capabilities of R1-Qwen-1.5B. Through a series of carefully designed experiments—spanning dataset curation, activation analysis, crosscoder training, feature patching, and scaling—I demonstrated that feature steering can be used to interpret and even improve model outputs. The findings not only offer insights into the internal mechanisms of chain-of-thought reasoning but also suggest that targeted interventions in latent space can lead to more robust and controllable generation.

Future Directions:

- Explore nonlinear transformations for feature steering to capture more complex interactions.
- Extend the analysis to other layers and models to generalize the findings.
- Integrate automated feature direction discovery (via PCA or supervised approaches) to refine our interventions further.