How Does Chain of Thought Think? Mechanistic Interpretability of Chain-of-Thought Reasoning with Sparse Autoencoding

Xi Chen, Aske Plaat, Niki van Stein
Leiden University
Einsteinweg 55, 2333 CC Leiden
The Netherlands
x.chen@liacs.leidenuniv.nl

Abstract

Chain-of-thought (CoT) prompting boosts Large Language Models accuracy on multi-step tasks, yet whether the generated "thoughts" reflect the true internal reasoning process is unresolved. We present the first feature-level causal study of CoT faithfulness. Combining sparse autoencoders with activation patching, we extract monosemantic features from Pythia-70M and Pythia-2.8B while they tackle GSM8K math problems under CoT and plain (noCoT) prompting. Swapping a small set of CoT-reasoning features into a noCoT run raises answer log-probabilities significantly in the 2.8B model, but has no reliable effect in 70M, revealing a clear contrast for these two scales. CoT also leads to significantly higher activation sparsity and feature interpretability scores in the larger model, signalling more modular internal computation. For example, the model's confidence in generating correct answers improves from 1.2 to 4.3. We introduce patch-curves and random-feature patching baselines, showing that useful CoT information is not only present in the top-K patches but widely distributed. Overall, our results indicate that CoT can induce more interpretable internal structures in high-capacity LLMs, validating its role as a structured prompting method.

Introduction

While Large Language Models (LLMs) have shown exceptional performance in reasoning tasks (Wei et al. 2022), their internal decision-making often remains a black box, making it hard for people to understand how the models reach their conclusions. In response to this challenge, mechanistic interpretability (MI) has emerged as a powerful alternative to traditional attributional methods (Chuang et al. 2024) and symbolic approaches (Xu et al. 2024; Li et al. 2024). Instead of relying on external proxies, MI investigates how specific features, neurons, or internal circuits contribute to reasoning. However, truly "looking inside" LLMs remains challenging: classic neuron-level analyses are limited by polysemanticity (Bricken et al. 2023) and superposition (Elhage et al. 2022), while circuit-level mapping often requires intensive manual effort, posing significant challenges for scaling to modern architectures (Nanda et al. 2023).

A promising approach in this area is the use of sparse autoencoders (SAEs) (Cunningham et al. 2023). By enforcing sparsity, SAEs help resolve polysemanticity and disentangle overlapping internal representations, producing monosemantic features that can be directly probed and causally manipulated. Furthermore, compared to component-level activation patching, which can be coarse-grained and ambiguous, feature-level interventions via SAEs provide potentially more targeted and semantically meaningful control over model behavior (Geiger et al. 2024; Marks et al. 2024).

Chain-of-Thought (CoT) prompting improves LLM performance on complex, multi-step reasoning tasks (Wei et al. 2022). However, it remains unclear whether CoT reasoning is faithful: whether the intermediate reasoning steps faithfully reflect the model's true internal decision-making process, or merely serve as plausible surface-level scaffolding. There is little feature-level, causally grounded analysis of reasoning faithfulness in LLMs, especially for math word problems requiring multi-step reasoning.

To address the question whether CoT enhances faithfulness of reasoning, we combine SAE and activation patching, to analyze the semantic features underlying LLM reasoning. By (1) training separate SAEs on CoT and NoCoT activations to extract dictionary features, and (2) performing a causal intervention by patching activations to swap these features between reasoning conditions. To further investigate the semantic alignment of these internal features, we also perform a lightweight interpretation that maps selected features to natural language descriptions. We go beyond attributional and symbolic methods to gain deeper insight into CoT reasoning. (Code at: https://github.com/sekirodie1000/cotFaithfulness). We make the following contributions (Figure 1):

- We introduce a feature-level causal intervention framework to mechanistically evaluate the faithfulness of CoT prompting in LLMs.
- We propose a log-probability-based evaluation procedure, enabling the systematic assessment of feature-level causal impacts in multi-step mathematical reasoning.
- We demonstrate on challenging math reasoning benchmarks that CoT induces sparser and more causally effective internal features, and thus indeed enhances faithful reasoning, but only in sufficiently large models.

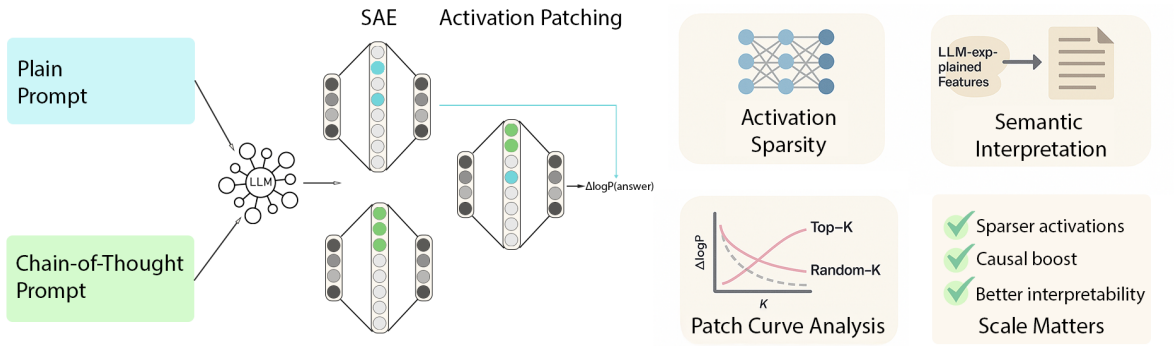

Figure 1: Workflow of the approach: After SAE, we do Activation patching, Feature Interpretation, and Activation Sparsity Analysis. All three confirm that CoT improves faithfulness of reasoning

Related Work

We present related work on (1) CoT, (2) SAE and interpretation, (3) faithfulness and causal interpretability.

Reasoning and CoT CoT prompting is effective at improving performance on complex tasks such as arithmetic and symbolic reasoning (Wei et al. 2022; Plaat et al. 2024). Zero-shot CoT shows that phrases like "Let's think step by step" can elicit coherent reasoning (Kojima et al. 2022). However, concerns remain: models sometimes reach correct answers despite incorrect intermediate steps (Yee et al. 2024), raising doubts about the faithfulness of CoT chains. We choose GSM8K as a challenging benchmark for evaluating CoT reasoning (Cobbe et al. 2021). GSM8K features complex problem structures, the answers often span multiple tokens, demanding high precision and logical coherence. If CoT truly reflects the model's internal problem-solving process, then the relevant causal features should still be identifiable even in these more challenging scenarios.

Sparse Autoencoder and Interpretation Recent work in MI uses SAEs to address superposition and polysemanticity in network representations (Cunningham et al. 2023). Max activation set analysis (Bills et al. 2023) and probing classifiers (Belinkov 2022) are limited by either scale or range of labels they use. By learning an overcomplete set of latent directions with a sparsity constraint, SAEs can break down dense model activations into monosemantic and interpretable features (Bricken et al. 2023; Braun et al. 2024). Crucially, SAE-derived features are not only interpretable but also causally manipulable. By intervening on these feature activations we can steer model behavior. Cunningham et al. (2023) used activation patching at the feature level and found that replacing or removing certain SAE features led to much larger changes in the model's outputs than PCA. Similarly, Bricken et al. (2023) used logit attribution to measure feature importance and showed that individual learned features make discernible contributions to the model. With SAE, interventions at the feature level, targeting meaningful and sparse features, and give more precise control of model behavior than changes made at the neuron or layer level.

Faithfulness and Causal Interpretability Faithfulness is defined as the degree to which an explanation reflects the model's true decision-making process (Agarwal, Tanneru, and Lakkaraju 2024). Several studies have proposed the use of counterfactual interventions to measure LLM faithfulness, such as CT/CCT frameworks (Atanasova et al. 2023; Siegel et al. 2024) and causal mediation analysis (Paul et al. 2024). They emphasize the causal relationship between model explanations and reasoning. Matton et al. (2025) introduce interventions at the semantic concept level, elevating faithfulness analysis to a higher abstraction level.

Causal analysis tools, such as activation patching (Meng et al. 2022), test the impact of interventions on internal activations. Unlike correlational measures they give direct evidence of causal influence. Interchange interventions (Geiger et al. 2021) and ablation find responsible components in model circuits, and recent small-scale studies were able to fully reverse-engineere transformer layers, confirming circuit functions by patching and ablation (Nanda et al. 2023).

Current approaches for evaluating the causality of CoT reasoning remain limited. Some studies attempt to intervene in the model's internal activation space to quantify the contribution of different parts of the reasoning chain to the final answer, making some progress in measuring faithfulness (Zhang and Nanda 2023; Yeo, Satapathy, and Cambria 2024). However, because reasoning in LLMs is highly parallel and redundant, most current interventions operate at the layer or component level, which makes it challenging to pinpoint the specific features or causal mechanisms responsible for model outputs. Backup circuits (a self-repair mechanism) further complicate attribution (Dutta et al. 2024).

To address these challenges, recent work has introduced feature-level interventions: interpretable feature directions are first extracted (e.g., via SAEs), and then directly manipulated. Geiger et al. (2024) argue that using learned feature subspaces enables finer tracking and control of model reasoning. Marks et al. (2024) further construct sparse causal circuits at the feature level, showing that a small number of key features can reconstruct complex behaviors. Makelov, Lange, and Nanda (2023) note that interventions in the feature subspace can sometimes lead to *interpretability illu-*

sions, where changes in model output not necessarily originate from the intended features. Wu et al. (2024) argue that this phenomenon reflects the inherent property of distributed representations in neural networks, and does not prevent patching or ablation methods from revealing effective structures in complex tasks. We further combine SAE feature space, activation patching, and CoT prompting to analyze the causal mechanisms in multi-step mathematical reasoning. By using methods such as Top-K patch curves, we provide a detailed characterization of the cumulative contribution of key features, advancing feature-level causal interpretation toward higher resolution and interpretability.

Our work goes beyond prior external and attributional approaches by directly probing LLM internal representations with mechanistic interpretability. We combine SAE-based feature extraction and activation patching to causally test whether CoT-elicited features enhance faithfulness, filling a key gap left by existing methods.

Methodology

To evaluate whether CoT improves the internal faithfulness of LLM reasoning, we use a feature-level causal analysis framework: (1) feature extraction, (2) causal intervention, (3) structural analysis, and (4) semantic interpretation. The framework uses SAEs to extract semantically meaningful sparse features from hidden states in the model. We then apply activation patching to exchange selected features between CoT and NoCoT conditions, allowing us to examine their causal impact on model outputs. To assess whether CoT prompts induce more structured computation, we compare activation sparsity across conditions. Finally, we generate natural language descriptions for SAE features and compute explanation scores to evaluate their semantic interpretability. This approach enables systematic, feature-level causal evaluation of reasoning faithfulness. We now describe the four components of our method.

Feature Extraction

For feature extraction, we use sparse autoencoders to extract salient latent features from the model's hidden representations $\mathbf{x} \in \mathbb{R}^{d_{input}}$ by learning a sparse dictionary of activation directions. Specifically, the SAE consists of an encoder $f_{enc}(\mathbf{x}) = \mathbf{h}$ that maps the high-dimensional activation $\mathbf{x}$ to a sparse feature vector $\mathbf{h} \in \mathbb{R}^k$, and a decoder $g_{dec}(\mathbf{h})$ that reconstructs the original input. To enforce sparsity, we include an L1 regularization term in the loss function. The total objective is:

$$L_{total} = L_{recon} + \lambda \|h\|_1$$

where $\mathcal{L}_{recon}$ is the reconstruction loss and λ controls the sparsity level.

We collect a large number of residual activations under both CoT and NoCoT prompting conditions, and train two separate SAE models to obtain distinct feature dictionaries $\mathbf{D}_{CoT}$ and $\mathbf{D}_{NoCoT}$. Each input $\mathbf{x}$ is encoded into a sparse vector $\mathbf{h}$, where the nonzero dimensions indicate which semantic features are activated and their respective strengths.

By extracting features with SAEs, we transform complex high-dimensional activations into a small number of latent features with clear semantic meaning. This approach builds on prior advances in neuron interpretability; for example, Cunningham et al. (2023) showed that training sparse dictionaries over activations can yield semantically meaningful features that support direct intervention. Extending this line of work, our study is the first to apply SAE-based feature extraction in the context of CoT prompting.

Causal Intervention

For causal intervention, we analyze the causal impact of features under different reasoning conditions using the activation patching method. While activation patching has been widely adopted in neural network interpretability research (Heimersheim and Nanda 2024), this work is the first to systematically integrate it with the SAE feature space to evaluate the faithfulness of CoT reasoning. By combining SAEs, activation patching, and CoT prompting, we construct a feature-level causal analysis framework that enables systematic evaluation and interpretation of reasoning faithfulness. This approach addresses limitations in prior work. Specifically, prior work either focused on neurons or layers (Dutta et al. 2024), or was limited to single-step reasoning (Hanna, Liu, and Variengien 2023). In contrast, our method provides a new tool for analyzing causal mechanisms in multi-step reasoning tasks. Concretely, for the same math problem prompted under both CoT and NoCoT conditions, we extract hidden activations $\mathbf{x}_{CoT}$ and $\mathbf{x}_{NoCoT}$, and obtain their sparse feature representations $\mathbf{h}_{CoT}$ and $\mathbf{h}_{NoCoT}$ using the SAE encoder. Given a feature subset S, we construct a patched feature vector by replacing the values of $\mathbf{h}_{NoCoT}$ with those from $\mathbf{h}_{CoT}$ on the selected subset:

$$\mathbf{h}_{patch}[S] = \mathbf{h}_{CoT}[S], \quad \mathbf{h}_{patch}[\bar{S}] = \mathbf{h}_{NoCoT}[\bar{S}].$$

This patched feature vector $\mathbf{h}_{patch}$ is then decoded back into activation space and forwarded through the remaining layers of the model to obtain a new output.

To quantify the causal effect of the patched features, we calculate the change in log-probability assigned to the correct answer before and after patching:

$$\Delta \log P = \log P_{patched}(answer) - \log P_{baseline}(answer).$$

A significant increase in confidence after inserting CoT features indicates that these features play a key causal role in the reasoning process.

To assess the cumulative effect of individual features, we perform patch curve analysis: features are ranked by the absolute difference $|\mathbf{h}_{CoT} - \mathbf{h}_{NoCoT}|$, and the Top-K features are gradually patched in. We compute $\Delta \log P$ at each step, yielding a curve that tracks how reasoning confidence changes as more features are introduced. To control for the selection bias of Top-K features, we also introduce a Random-K baseline, where K features are randomly sampled from the full feature set at each step for patching. By comparing the patch curves of Top-K and Random-K, we assess whether the causal effects are concentrated in the most differentiated features or distributed more broadly.

Although both activation patching and SAEs are existing tools, this is the first study to combine them for analyzing CoT reasoning faithfulness in mathematical tasks.

Prior work often operated in raw activation or neuron space, where overlapping and polysemantic representations make interpretation difficult (Marks et al. 2024). By applying activation patching in the SAE-derived feature space, we enable higher-resolution and more semantically targeted interventions. Together with log-probability-based evaluation, this framework provides a precise, interpretable method for assessing reasoning faithfulness in complex multi-step reasoning tasks.

Structural Analysis

For structural analysis, we quantify activation sparsity to compare the internal computation focus under CoT and NoCoT conditions. Activation sparsity measures the proportion of units in a model's hidden state that are inactive (close to zero) for a given input.

Let $x^{(l)} \in \mathbb{R}^{T \times d}$ denote the activations at layer l for a sequence of length T and hidden size d. The global sparsity for a threshold ϵ is:

$$Sparsity(x^{(l)}) = 1 - \frac{1}{T \cdot d} \sum_{t=1}^{T} \sum_{j=1}^{d} \mathbb{I}\left[|x_{t,j}^{(l)}| > \epsilon\right]$$

where $I[\cdot]$ is the indicator function that returns 1 if its argument is true and 0 otherwise, and ϵ is a small positive threshold.

To enable efficient computation, especially for large models, we divide the sequence into N non-overlapping chunks of size $C = T/N$. For the i-th chunk, the chunk-wise sparsity is defined as:

$$ChunkSparsity_i = 1 - \frac{1}{n \cdot d} \sum_{t \in chunk_i} \sum_{j=1}^{d} \mathbb{I}\left[|x_{t,j}^{(l)}| > \epsilon\right]$$

Here, $chunk_i$ refers to the set of time steps belonging to the i-th chunk, with $n = |chunk_i|$ denoting its length.

After calculating sparsity for all chunks, we aggregate these results to obtain the global sparsity distribution across the entire dataset. This chunk-based computation is purely technical, enabling efficient processing without changing the underlying global sparsity definition. To our knowledge, while sparse activations are often considered a signal of improved interpretability and modularity (Cunningham et al. 2023), few studies have examined this in the context of CoT reasoning. Through a systematic comparison of activation sparsity under CoT and NoCoT prompting, our study is the first to reveal how prompting strategies influence the internal activation structure of the model—offering important insights into how CoT prompts reshape internal computation.

Semantic Interpretation

For semantic interpretation, we assign each SAE feature an interpretable explanation by collecting highly activating text snippets and using a language model to generate and simulate natural language descriptions (as in (Bills et al. 2023)). The explanation's quality is evaluated by correlating predicted and true activation sequences, yielding an interpretation score. We compare the distribution of interpretation scores under CoT and NoCoT, using both statistical tests and box plots.

In our framework, the semantic interpretation module builds on prior work that uses LLMs to automatically generate feature-level semantic labels. However, we apply this technique to a novel comparative setting, analyzing differences in semantic consistency between internal features under CoT and NoCoT prompting. This perspective has not been systematically explored before. By combining explanation scores with results from explanation scores, causal patching, and activation sparsity, we gain a more comprehensive view of whether CoT prompts guide the model to learn more meaningful intermediate representations, thereby enabling a systematic evaluation of CoT faithfulness.

Experiment Setup

We selected two pretrained language models released by EleutherAI, Pythia-70M (6 layers, 512 hidden, 8 heads, FFN size 2048) and Pythia-2.8B (32 layers, 2560 hidden, 32 heads, FFN size 10240), both trained on the Pile and using the same vocabulary and tokenizer. We used the public Pythia v0 weights and performed only post-hoc analysis.

As our benchmarks we used GSM8K, containing grade-school level math word problems. All analyses were conducted on the training split. Two input formats were used: CoT (three fixed few-shot examples, each with detailed step-by-step solutions) and NoCoT (only the current problem). Prompts were hardcoded and identical across the dataset. Only the question was used as model input, with no ground-truth answer provided during inference; ground-truth was used only for evaluation. To avoid bias, both formats were processed using the same pipeline, with a max input length of 256 tokens. Activations were extracted from the residual stream of layer 2 at the final token position, as (Cunningham et al. 2023) shows that features learned by sparse autoencoders are significantly more interpretable in the early layers of the residual stream, with interpretability scores dropping in deeper layers. For both models, the training data for SAEs under CoT and NoCoT was identical except for the input format. SAEs were trained separately for each model and prompt setting, with dictionary ratios of 4 and 8 representing lower and higher sparsity. Multiple SAE variants were trained per model/layer, with a representative subset chosen for downstream analysis.

For activation patching, we used two feature selection schemes:

1. **Top-K**: the K sparse features with the largest absolute activation difference $|h_A^{(l)} - h_B^{(l)}|$.
2. **Random-K**: a control variant that patches K features uniformly sampled from the full dictionary.

For distributional analyses, we fix $K = 20$. For patch-curve experiments, we vary $K \in 1, 2, 4, 8, 16, 32, 64, 128, 256$, capping at 256 features. Up to 1000 problem pairs per condition were evaluated. All model operations used HuggingFace Transformers and TransformerLens. Feature interpretation was performed using GPT-3.5-turbo on top-activating contexts. We used a NVIDIA A100 GPU, 18 CPU cores, and 90GB of RAM.

Full implementation and definitions are provided in the Appendix (Chen, Plaat, and van Stein 2025).

Results

We analyze CoT and NoCoT reasoning in LLMs on the GSM8K dataset, evaluating feature interpretability, causal influence, and activation sparsity.

Effect of CoT on Feature Interpretability

We first compared the explanation scores of features learned under CoT versus NoCoT prompting. Figure 2 shows the score distributions for Pythia-70M and Pythia-2.8B under a dictionary sparsity ratio of 4. Table 1 summarizes the corresponding statistical results. For Pythia-70M, the average explanation score under CoT was 0.018, compared to 0.016 under NoCoT, a slight improvement. The box plot in Figure 2 further shows that features under NoCoT performed slightly better in terms of interpretability: the median score is higher and outliers are more positive. A t-test confirms this, yielding a t-value of 0.082 and a p-value of 0.935, suggesting that CoT may slightly hinder interpretability in smaller models. For Pythia-2.8B, the average explanation score under CoT was 0.056, higher than –0.013 for NoCoT. As shown in Figure 2, features activated by CoT prompts display a broader distribution, with some reaching values around 0.6. This suggests that CoT elicits semantically coherent internal features in larger models. The t-test shows this difference is statistically significant ($t = 2.96$, $p = 0.004$).

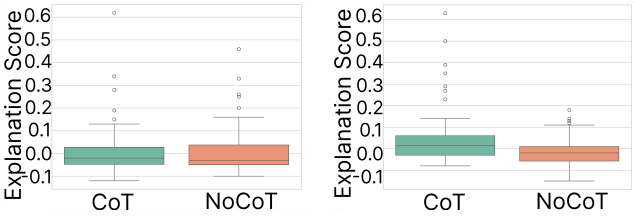

Figure 2: Comparison of feature explanation scores under CoT and NoCoT prompts. Left: Pythia-70M; Right: Pythia-2.8B. The 2.8B model shows higher explanation scores under CoT, indicating stronger causal features are learned in the larger model when CoT prompting is applied. Each plot is based on 50 features per condition.

In summary, while CoT is not sufficient for logically faithful reasoning chains in LLMs, it serves as an effective structural prompt in larger models, nudging them toward more semantically coherent internal features. In smaller models, the effect remains minimal. These findings are consistent with our activation patching experiments, where CoT-elicited features in larger models demonstrated causal influence on output behavior.

Causal Effects of CoT Features via Activation Patching

We next examine the causal role of learned sparse features through controlled activation patching. Specifically, we inject the top-K most salient sparse features from a CoT forward pass into a NoCoT pass, and vice versa, to assess their impact on output log-probabilities for the correct answer.

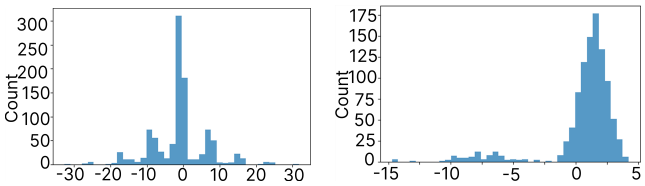

Figure 3: Distribution of log-probability changes after patching the top 20 CoT features into NoCoT runs under dictionary ratio 4. Left: Pythia-70M; Right: Pythia-2.8B. While 2.8B shows a strong positive shift indicating consistent benefit from CoT features, 70M shows highly variable effects, including large performance drops, suggesting unstable or less effective feature transfer.

In Pythia-2.8B, CoT-to-NoCoT patching consistently improves performance, while NoCoT-to-CoT patching has minimal effect. Figures 3 and 4 show that log-probability deltas after CoT patching are predominantly positive. In contrast, the same patching in Pythia-70M yields highly variable, often symmetric distributions, with both large gains and losses, indicating that CoT features do not reliably transfer in the smaller model and can disrupt original inference.

When varying the number of patched features K, the patching curves (Figures 5 and 6) reveal that in Pythia-2.8B, injecting CoT features immediately yields a strong gain that gradually saturates, while in Pythia-70M, CoT patching leads to no gain or even performance drops. Notably, under higher sparsity (dictionary ratio = 8), these trends are even more pronounced: Pythia-2.8B's CoT curve exceeds +3.2 at K=2, then stabilizes; Pythia-70M shows persistent declines, indicating CoT features do not provide robust benefit in small models.

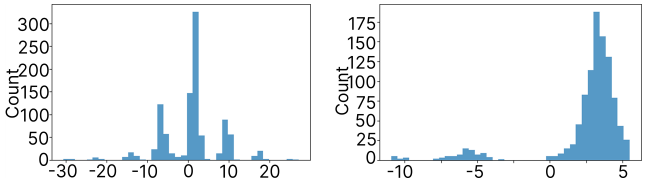

Figure 4: Distribution of log-probability changes after patching the top 20 CoT features into NoCoT runs under dictionary ratio 8. Left: Pythia-70M; Right: Pythia-2.8B. Compared to ratio 4, the distributions are similar: 2.8B continues to show consistent improvements, while 70M remains less robust, exhibiting high variance and frequent negative effects.

Crucially, random-K controlled experiments reveal that, in Pythia-2.8B, randomly sampling K CoT-activated features often outperforms selecting the Top-K by activation. For example, the model's confidence in generating correct answers improves from 1.2 to 4.3. This suggests that useful information from CoT prompts is widely distributed among moderately activated features, rather than concentrated in a

Model	CoT Mean	CoT Std	NoCoT Mean	NoCoT Std	t-stat	p-value
Pythia-70M	0.018	0.125	0.016	0.116	0.082	0.935
Pythia-2.8B	0.056	0.147	-0.013	0.071	2.96	0.004

Table 1: Statistical comparison of feature explanation scores under CoT and NoCoT prompts. Results are shown for Pythia-70M and Pythia-2.8B, including mean, standard deviation, and T-test statistics.

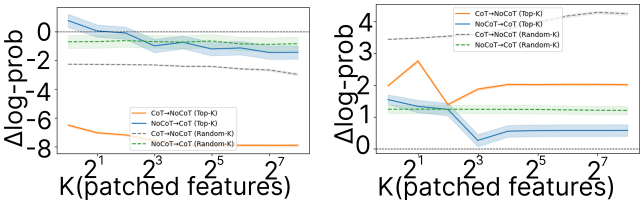

Figure 5: Top-K and Random-K patching performance under dictionary ratio 4. Left: Pythia-70M; Right: Pythia-2.8B. CoT→NoCoT patching shows the effect of patching CoT features into NoCoT, while NoCoT→CoT patching shows the reverse. In 2.8B, patching CoT features yields consistent performance gains, highlighting their causal importance. In contrast, for 70M, patching CoT features leads to a substantial and monotonic performance decline, suggesting that CoT-induced features are ineffective or even harmful in the smaller model ($p < 0.001$).

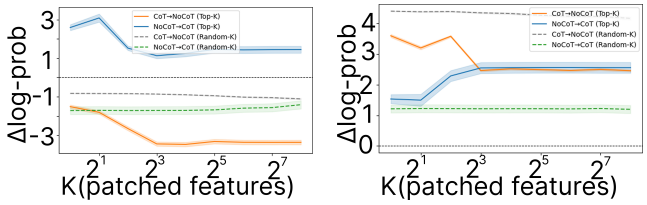

Figure 6: Top-K and Random-K patching performance under dictionary ratio 8. Left: Pythia-70M; Right: Pythia-2.8B. For 2.8B, CoT→NoCoT patching consistently improves performance, with diminishing returns as K increases. NoCoT→CoT patching gradually degrades the CoT run, suggesting CoT features are causally significant and sparse. In contrast, for 70M, patching CoT features into NoCoT runs still causes a net performance drop, though less sharply than under ratio 4. Interestingly, NoCoT→CoT patching shows mild improvement ($p < 0.001$).

few top directions. The Top-K strategy may overfit to local peaks, missing supportive features that random selection includes, resulting in more stable and comprehensive positive effects. This distributed effect is not observed in Pythia-70M, where both random and Top-K patching fail to consistently improve performance. This suggests that in large models, the causal signal from CoT is not limited to the most activated features, but is spread across many, making random selection more effective than simply taking the strongest activations. The next section further explains this phenomenon by analyzing the structure and sparsity of feature activations.

Activation Sparsity under CoT and NoCoT

Following the causal intervention experiments, we turn to the structural properties of internal activations. We focus on sparsity—how CoT and NoCoT prompts affect the distribution and density of activated neurons and SAE features across model sizes. Figure 7 shows that CoT prompts lead to sparser residual activations. In the NoCoT condition, more neurons exhibit moderate/high activation; under CoT, most neurons are near zero, with few strong activations. This effect is more pronounced in the 2.8B model, where CoT activations are almost entirely low except for a small subset.

To further analyze this, we use SAE to extract feature representations and count the number of activated neurons per SAE feature. Figures 8 and 9 show that under CoT, each SAE feature tends to activate only a small number of neurons, while under NoCoT, features often activate a broader set. In the 2.8B model, many CoT features are supported by only a handful of neurons, indicating a more pronounced structured sparsity. Interestingly, this structured sparsity in CoT-induced representations also helps explain the surprising result from our patching experiments: in the 2.8B model, randomly sampled CoT features consistently outperform top-ranked ones when patched into NoCoT trajectories. At first glance, this seems counterintuitive—why would random features yield better performance than those with the highest activation? As shown earlier, CoT prompting not only increases global activation sparsity, but also leads to higher feature-level variability in the large model. Under CoT, most neurons have their activations suppressed close to zero, with only a small number strongly activated, and the number of neurons involved in different features varies greatly. This "structured sparsity" means that the useful information activated by CoT prompts is not concentrated in a few highly activated features, but is more widely spread across many moderately activated ones. The Top-K strategy may overfit to local peaks and miss supportive features, while random sampling is more likely to include these overlooked features, leading to more stable and comprehensive positive effects in patching.

Overall, these results show that CoT prompting not only improves reasoning performance but also reshapes the model's internal activation patterns. In both 70M and 2.8B, CoT leads to fewer neurons being activated overall, especially in the large model. At the SAE feature level, there is greater variation in how many neurons are engaged by each feature, suggesting that CoT encourages semantic resource allocation and latent disentanglement. This trend is especially prominent in 2.8B, enabling random feature patching to be surprisingly effective.

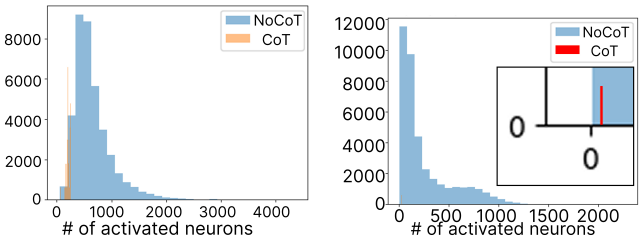

Figure 7: Sparsity comparison of residual activations under CoT and NoCoT prompts. Left: Pythia-70M; Right: Pythia-2.8B. In both models, CoT leads to significantly sparser residual activations, with most neurons remaining near zero and only a small subset strongly activated. This sparsity effect is markedly more pronounced in the 2.8B model, indicating enhanced activation selectivity at larger scale.

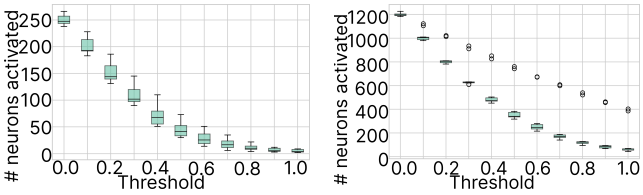

Figure 8: Activated neuron counts per SAE feature under NoCoT prompting, across thresholds from 0.0 to 1.0. Left: Pythia-70M; Right: Pythia-2.8B. The large model (2.8B) activates significantly more neurons per feature at each threshold, indicating denser feature composition compared to the small model.

Discussion

Using MI, we investigate whether CoT prompting improves the faithfulness of the reasoning processes within LLMs. Our analysis address the following three research questions:

First we studied if CoT encourages the model to learn internal features that are more semantically consistent and easier to interpret. CoT prompts substantially indeed improve the semantic coherence and interpretability of internal features, but only in the larger model. In Pythia-2.8B, features learned under CoT display higher explanation scores and semantic consistency; in 70M the effect is small.

Next we analyzed with activation patching if CoT enhances the causal relevance of internal features. Activation patching experiments reveal a clear scale-dependent effect: in the large model, injecting random sets of CoT features into NoCoT forward passes significantly boosts output log-probabilities, demonstrating a strong causal influence. In contrast, similar interventions in the small model fail to improve, and sometimes even degrade, performance.

Finally we studied if CoT can promote sparser feature activations, a property commonly associated with enhanced interpretability. CoT prompts induce much sparser activation patterns, especially in the 2.8B model, where both residual stream and SAE feature activations are suppressed except for a small subset. This structured sparsity enables more focused semantic allocation and explains the effectiveness of random feature patching.

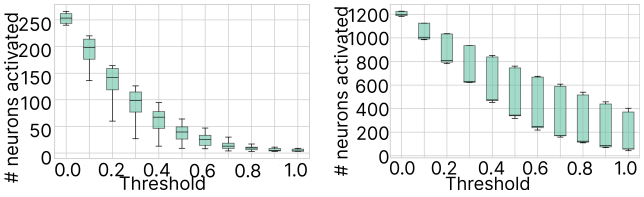

Figure 9: Activated neuron counts per SAE feature under CoT prompting. Left: Pythia-70M; Right: Pythia-2.8B. Compared to NoCoT, CoT prompts yield substantially sparser activations in both models, with 2.8B showing stronger sparsity and higher inter-feature variance.

Limitation and Future Work

This study is limited in several aspects. First, our activation patching targets only the residual activation of the final token and does not trace causal effects through the reasoning process; this is fundamentally due to the static, snapshot-based nature of the SAE framework, which is incompatible with token-level or path-level causal tracing methods (Goldowsky-Dill et al. 2023; Zhang and Nanda 2023). Second, our interpretation module relies on OpenAI's LLM-based scoring (Agarwal, Tanneru, and Lakkaraju 2024), which offers an indirect perspective and does not ground explanations in specific neurons or heads, nor validate them with causal interventions (Geiger et al. 2023). Third, experiments are restricted to Pythia-2.8B and smaller variants; we did not include larger models such as LLaMA-7B, and our findings may not generalize (Shojaee et al. 2025; Demircan et al. 2024). Fourth, SAE-based feature analysis introduces biases and may miss distributed or entangled representations (Dooms and Wilhelm 2025; Karvonen et al. 2024; Bereska and Gavves 2024). Not all interpretable SAE features have causal effects (Menon et al. 2024).

For future work, we suggest conducting token-level and path-based causal analysis, ideally in combination with SAE-based feature decomposition, such as stepwise interventions and path patching (Goldowsky-Dill et al. 2023). It is important to develop activation-grounded and causally-validated explanation methods, including probing, clustering, and patching (Geiger et al. 2023; Tighidet et al. 2024; Bills et al. 2023). Further research should scale this framework to larger models and diverse architectures, and explore subspace patching and automated circuit discovery tools for more precise mechanistic analysis.

Conclusion

This study combined sparse autoencoding, activation patching, and automated feature interpretation to probe the internal faithfulness of CoT reasoning in LLMs. Our findings show that CoT prompts, especially in the larger Pythia model, induce more semantically coherent, causally effective, and sparser internal features. However, these effects are minimal in the smaller 70M model. This work highlights how CoT not only improves performance but also reshapes internal reasoning processes, offering new insight into the mechanisms underlying LLM reasoning.

Acknowledgments

This work was partially funded by the OTP research project "SuperCode: SUstainability PER AI-driven CO-DEsign", project number 2025/TTW/01878134, which is financed by the Dutch Research Council (NWO) domain Applied and Engineering Sciences (TTW).

References

Agarwal, C.; Tanneru, S. H.; and Lakkaraju, H. 2024. Faithfulness vs. plausibility: On the (un) reliability of explanations from large language models. *arXiv preprint arXiv:2402.04614*.

Atanasova, P.; Camburu, O.-M.; Lioma, C.; Lukasiewicz, T.; Simonsen, J. G.; and Augenstein, I. 2023. Faithfulness tests for natural language explanations. *arXiv preprint arXiv:2305.18029*.

Belinkov, Y. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1): 207–219.

Bereska, L.; and Gavves, E. 2024. Mechanistic Interpretability for AI Safety–A Review. *arXiv preprint arXiv:2404.14082*.

Bills, S.; Cammarata, N.; Mossing, D.; Tillman, H.; Gao, L.; Goh, G.; Sutskever, I.; Leike, J.; Wu, J.; and Saunders, W. 2023. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Braun, D.; Taylor, J.; Goldowsky-Dill, N.; and Sharkey, L. 2024. Identifying functionally important features with end-to-end sparse dictionary learning. *Advances in Neural Information Processing Systems*, 37: 107286–107325.

Bricken, T.; Templeton, A.; Batson, J.; Chen, B.; Jermyn, A.; Conerly, T.; Turner, N.; Anil, C.; Denison, C.; Askell, A.; et al. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2.

Chen, X.; Plaat, A.; and van Stein, N. 2025. How does chain of thought think? mechanistic interpretability of chain-of-thought reasoning with sparse autoencoding. *arXiv preprint arXiv:2507.22928*.

Chuang, Y.-N.; Wang, G.; Chang, C.-Y.; Tang, R.; Zhong, S.; Yang, F.; Du, M.; Cai, X.; and Hu, X. 2024. FaithLM: Towards faithful explanations for large language models. *arXiv preprint arXiv:2402.04678*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Cunningham, H.; Ewart, A.; Riggs, L.; Huben, R.; and Sharkey, L. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Demircan, C.; Saanum, T.; Jagadish, A. K.; Binz, M.; and Schulz, E. 2024. Sparse autoencoders reveal temporal difference learning in large language models. *arXiv preprint arXiv:2410.01280*.

Dooms, T.; and Wilhelm, D. 2025. Tokenized SAEs: Disentangling SAE Reconstructions. *arXiv preprint arXiv:2502.17332*.

Dutta, S.; Singh, J.; Chakrabarti, S.; and Chakraborty, T. 2024. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *arXiv preprint arXiv:2402.18312*.

Elhage, N.; Hume, T.; Olsson, C.; Schiefer, N.; Henighan, T.; Kravec, S.; Hatfield-Dodds, Z.; Lasenby, R.; Drain, D.; Chen, C.; et al. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Geiger, A.; Ibeling, D.; Zur, A.; Chaudhary, M.; Chauhan, S.; Huang, J.; Arora, A.; Wu, Z.; Goodman, N.; Potts, C.; et al. 2023. Causal abstraction: A theoretical foundation for mechanistic interpretability. *arXiv preprint arXiv:2301.04709*.

Geiger, A.; Lu, H.; Icard, T.; and Potts, C. 2021. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34: 9574–9586.

Geiger, A.; Wu, Z.; Potts, C.; Icard, T.; and Goodman, N. 2024. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, 160–187. PMLR.

Goldowsky-Dill, N.; MacLeod, C.; Sato, L.; and Arora, A. 2023. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*.

Hanna, M.; Liu, O.; and Variengien, A. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36: 76033–76060.

Heimersheim, S.; and Nanda, N. 2024. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*.

Karvonen, A.; Rager, C.; Marks, S.; and Nanda, N. 2024. Evaluating Sparse Autoencoders on Targeted Concept Erasure Tasks. *arXiv preprint arXiv:2411.18895*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213.

Li, Q.; Li, J.; Liu, T.; Zeng, Y.; Cheng, M.; Huang, W.; and Liu, Q. 2024. Leveraging LLMs for Hypothetical Deduction in Logical Inference: A Neuro-Symbolic Approach. *arXiv preprint arXiv:2410.21779*.

Makelov, A.; Lange, G.; and Nanda, N. 2023. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. *arXiv preprint arXiv:2311.17030*.

Marks, S.; Rager, C.; Michaud, E. J.; Belinkov, Y.; Bau, D.; and Mueller, A. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*.

Matton, K.; Ness, R. O.; Guttag, J.; and Kıcıman, E. 2025. Walk the talk? Measuring the faithfulness of large language model explanations. *arXiv preprint arXiv:2504.14150*.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35: 17359–17372.

Menon, A.; Shrivastava, M.; Krueger, D.; and Lubana, E. S. 2024. Analyzing (In) Abilities of SAEs via Formal Languages. *arXiv preprint arXiv:2410.11767*.

Nanda, N.; Chan, L.; Lieberum, T.; Smith, J.; and Steinhardt, J. 2023. Progress measures for grokking via mechanistic interpretability, 2023. *URL https://arxiv. org/abs/2301.05217*.

Paul, D.; West, R.; Bosselut, A.; and Faltings, B. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. *arXiv preprint arXiv:2402.13950*.

Plaat, A.; Wong, A.; Verberne, S.; Broekens, J.; van Stein, N.; and Back, T. 2024. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*.

Shojaee, P.; Mirzadeh, I.; Alizadeh, K.; Horton, M.; Bengio, S.; and Farajtabar, M. 2025. The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity.

Siegel, N. Y.; Camburu, O.-M.; Heess, N.; and Perez-Ortiz, M. 2024. The probabilities also matter: A more faithful metric for faithfulness of free-text explanations in large language models. *arXiv preprint arXiv:2404.03189*.

Tighidet, Z.; Mogini, A.; Mei, J.; Piwowarski, B.; and Gallinari, P. 2024. Probing Language Models on Their Knowledge Source. *arXiv preprint arXiv:2410.05817*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wu, Z.; Geiger, A.; Huang, J.; Arora, A.; Icard, T.; Potts, C.; and Goodman, N. D. 2024. A reply to makelov et al.(2023)'s" interpretability illusion" arguments. *arXiv preprint arXiv:2401.12631*.

Xu, J.; Fei, H.; Pan, L.; Liu, Q.; Lee, M.-L.; and Hsu, W. 2024. Faithful logical reasoning via symbolic chain-of-thought. *arXiv preprint arXiv:2405.18357*.

Yee, E.; Li, A.; Tang, C.; Jung, Y. H.; Paturi, R.; and Bergen, L. 2024. Dissociation of faithful and unfaithful reasoning in llms. *arXiv preprint arXiv:2405.15092*.

Yeo, W. J.; Satapathy, R.; and Cambria, E. 2024. Towards faithful natural language explanations: A study using activation patching in large language models. *arXiv preprint arXiv:2410.14155*.

Zhang, F.; and Nanda, N. 2023. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*.