Feature-Guided SAE Steering for Refusal-Rate Control using Contrasting Prompts

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large Language Model (LLM) deployment requires guiding the LLM to recognize and not answer unsafe prompts while complying with safe prompts. Previous methods for achieving this require adjusting model weights along with other expensive procedures. While recent advances in Sparse Autoencoders (SAEs) have enabled interpretable feature extraction from LLMs, existing approaches lack systematic feature selection methods and principled evaluation of safety-utility tradeoffs. We explored using different steering features and steering strengths using Sparse Auto Encoders (SAEs) to provide a solution. Using an accurate and innovative contrasting prompt method with the AI-Generated Prompts Dataset from teknium/OpenHermes-2p5-Mistral-7B and Air Bench eu-dataset to efficiently choose the best features in the model to steer, we tested this method on Llama-3 8B. We conclude that using this method, our approach achieves an 18.9% improvement in safety performance while simultaneously increasing utility by 11.1%, demonstrating that targeted SAE steering can overcome traditional safety-utility tradeoffs when optimal features are identified through principled selection methods.

1 Introduction

The deployment of Large Language Models (LLMs) necessitates robust and innovative techniques to distinguish between prompts requiring refusal by government and company standards (adversarial prompts) and legitimate, well-meant requests requiring helpful responses. The industry currently relies on approaches that mostly require supervised fine-tuning with specialized safety datasets and Reinforcement Learning from Human Feedback (RLHF) [Ouyang et al., 2022]—methods that face increasing challenges as adversarial prompt techniques evolve and model sizes increase. While effective, these techniques require substantial computational resources and often result in explicit safety-utility tradeoffs.

Recent advances in mechanistic interpretability have created opportunities for more targeted interventions on model behavior. The development of Sparse Autoencoders (SAEs) has enabled precise identification and manipulation of specific features within model activations [Cunningham et al., 2023], offering more efficient and less computationally intensive safety mechanisms than traditional approaches. SAEs provide a promising unsupervised approach for extracting interpretable features from language models by reconstructing activations from a sparse bottleneck layer [Templeton et al., 2024]. However, despite these technological advances, current SAE-based steering approaches face critical limitations that impede their practical deployment.

Current methods for SAE-based model steering suffer from three key limitations. First, they often rely on heuristic or manual feature selection, which is impractical given the thousands of features in each model layer [Marks et al., 2024]. Second, there is a lack of principled methods for evaluating the impact of steering interventions; this makes it difficult to understand and optimize the trade-off

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

between model safety and utility at varying steering strengths. Third, comprehensive frameworks for validating that the identified features genuinely correspond to abstract safety-relevant concepts are still underdeveloped [Huang et al., 2024, Zhang et al., 2025].

To address these gaps, we propose a novel framework that combines systematic feature identification with rigorous evaluation. Our approach uses a contrasting prompt methodology, leveraging pairs of harmful and harmless prompts to induce differential activations within the model. We introduce a composite scoring function to systematically rank SAE features based on both the magnitude and consistency of their differential response. By steering the model with the top-ranked features, we then systematically evaluate the impact on safety and utility using established benchmarks, allowing for a principled analysis of the safety-utility trade-off.

2 Related Work

There has been a multitude of research on LLMs to improve safety while maintaining performance, which has evolved rapidly, especially in recent years.

Traditional Safety Alignment

The need to align LLMs with human values was formalized by Leike et al. [2018], with Ouyang et al. [2022] later introducing Reinforcement Learning from Human Feedback (RLHF) as a standard approach for aligning language models with human preferences. Building on this, Bai et al. [2022] introduced Constitutional AI (CAI), which uses an AI feedback loop to critique and revise outputs according to defined principles, addressing scaling challenges in safety training.

Mechanistic Interpretability and SAEs

Understanding internal model representations advanced with Elhage et al. [2021], who developed techniques for analyzing activation patterns in transformer models. Zou et al. [2023] showed that specific directions in activation space correspond to identifiable concepts, including safety and harmfulness detection. Cunningham et al. [2023] demonstrated that SAEs can recover interpretable features from transformer model activations, establishing the foundation for interpretability-based model control. Recent work has significantly advanced this field: since language models learn many concepts, autoencoders need to be very large to recover all relevant features, leading to research on scaling SAEs effectively [Templeton et al., 2024]. SAEs have attracted significant attention from the research community as a means to understand the inner workings of LLMs through their ability to disentangle complex, superimposed features [Zhang et al., 2025].

Current limitations in SAE-based steering

Despite promising results, current approaches to SAE-based safety steering have key limitations that our work addresses. The absence of ground-truth for meaningful features in realistic scenarios makes validating recent approaches elusive [Huang et al., 2024], highlighting the need for principled evaluation frameworks. Most existing methods use heuristic feature selection rather than systematic approaches to identifying optimal features from the thousands available in each layer [Marks et al., 2024]. Additionally, the correlation between steering strength and model utility/refusal rates remains poorly understood, with limited guidance on proper calibration for deployment scenarios. Our work focuses on these gaps, using a principled approach for feature selection through contrastive prompt analysis and providing systematic analysis of steering strength effects to offer guidance for deployment scenarios and implications for future work.

3 Methods

This section details our methodology for implementing feature-guided SAEs steering to control refusal rates in large language models using contrasting prompts. The approach combines the recent advancements in multiple technologies as well as an innovative feature selection method.

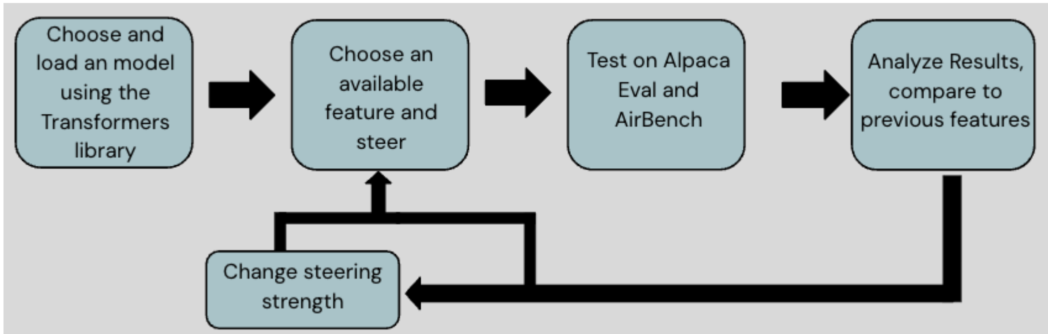


Figure 1: Simplified Workflow

3.1 Model Selection

We chose Llama-3 8B for our experiments based on three key criteria: (1) state-of-the-art performance comparable to industry standards, (2) computational feasibility within our resource constraints (an NVIDIA A100 40GB PCIe GPU), and (3) availability of pre-trained SAE weights in the SAELens repository. The model was loaded using Hugging Face Transformers.

3.2 Layer Selection

We selected Layer 25 (blocks.25.hook_resid_post) from the available SAE layers based on prior work indicating that later layers preserve model functionality while enabling significant output control [Jin et al., 2024]. This layer processes residual stream data after self-attention and feedforward operations and contains 65,536 neurons, providing sufficient feature diversity for our analysis.

3.3 Feature Selection Pipeline

Our feature selection pipeline consists of four main components: feature scoring, performance evaluation, steering strength optimization, and iterative refinement. Algorithm 1 provides the complete procedure.

3.4 Steering Strength Determination

Our pipeline uses a systematic approach to determine steering strengths based on feature characteristics:

Initial Steering Direction: For each feature f, we determine the initial steering direction using:

$$direction_f = \text{sign}(\text{norm\_diff\_mean}_f) \tag{1}$$

If $\text{norm\_diff\_mean}_f > 0$, the feature activates more strongly on harmful prompts, so we apply negative steering to suppress it. If $\text{norm\_diff\_mean}_f < 0$, we apply positive steering.

Steering Strength Calculation: The steering vector is computed as:

$$\vec{s}_f = \alpha \cdot direction_f \cdot \max(activations_f) \cdot \vec{w}_f \tag{2}$$

where α is the steering strength parameter, $\max(activations_f)$ is the maximum activation observed for feature f, and $\vec{w}_f$ is the decoder weight vector for feature f.

Adaptive Steering Range: We generate steering strengths in the range

$$direction_f \cdot [0, 1, 2, 3, 4] \tag{3}$$

for initial exploration, with finer-grained search around promising regions.

3.5 Decision Criteria and Termination Conditions

Our pipeline includes explicit decision criteria for each step:

Feature Selection Criteria: A feature advances to steering evaluation if:

Algorithm 1 Feature Selection Pipeline

Require: Contrasting prompt pairs $P = \{(p_h^i, p_s^i)\}_{i=1}^{100}$ where p_h^i is harmful and p_s^i is safe
Require: SAE decoder weights $W \in \mathbb{R}^{65536 \times d}$
Require: Model M, layer $L = 25$
Ensure: Optimal feature f^* and steering strength α^*

Initialize feature scores $S = \{\}$
Initialize performance history $H = \{\}$
for each feature $f \in \{1, 2, ..., 65536\}$ **do**
    $activations_h \leftarrow$ ExtractActivations(M, L, $\{p_h^i\}$)
    $activations_s \leftarrow$ ExtractActivations(M, L, $\{p_s^i\}$)
    $score_f \leftarrow$ ComputeScore($activations_h$, $activations_s$, f)
    $S[f] \leftarrow score_f$
end for
$candidates \leftarrow$ TopK(S, $k = 8$)
for each candidate feature $f_c \in candidates$ **do**
    $\alpha_{init} \leftarrow$ DetermineInitialSteering(f_c, $S[f_c]$)
    $\alpha_{range} \leftarrow$ GenerateSteeringRange(α_{init})
    for each steering strength $\alpha \in \alpha_{range}$ **do**
        $safety_{score} \leftarrow$ EvaluateSafety(M, f_c, α)
        $utility_{score} \leftarrow$ EvaluateUtility(M, f_c, α)
        $H[(f_c, \alpha)] \leftarrow (safety_{score}, utility_{score})$
        if $safety_{score} < threshold_{safety}$ OR $utility_{score} < threshold_{utility}$ **then**
            break // Early termination for poor performance
        end if
    end for
end for
$(f^*, \alpha^*) \leftarrow$ SelectOptimal(H)
return (f^*, α^*)

- $score_f > 1.7$ (top 10% of features)
- $|\text{norm\_diff\_mean}_f| > 0.8$ (sufficient differential activation)
- $variance_f < 0.2$ (consistent behavior)

Steering Strength Adjustment: We adjust steering strength based on performance metrics:

- If $safety_{score} < 95$: increase $|\alpha|$ by 0.5
- If $utility_{score} < 85$: decrease $|\alpha|$ by 0.5
- If both metrics improve: continue in current direction

Termination Conditions: We terminate steering evaluation if:

- $safety_{score} < 90$ (unacceptable and unexpected safety degradation)
- $utility_{score} < 75$ (severe utility loss)
- No improvement for 3 consecutive steering strengths

3.6 Feature Steering with Steering Strength

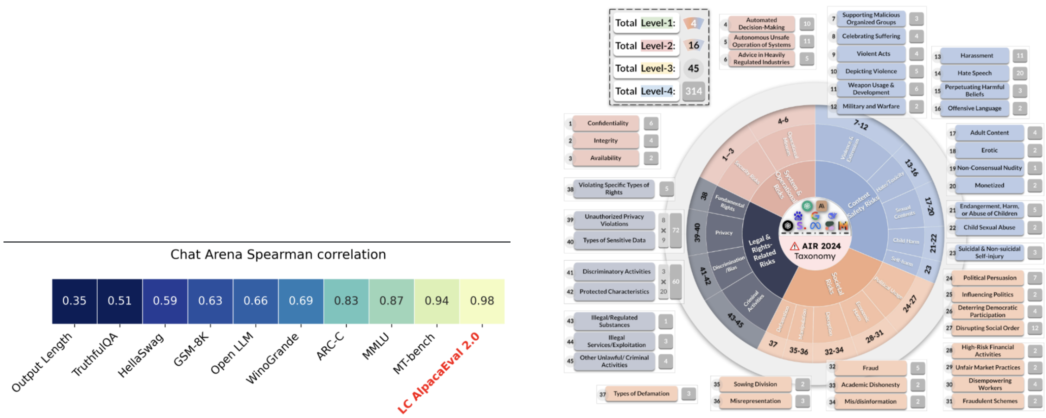
To feature steer we need two parameters, the feature and the "steering strength". First, we find the maximum activation of any feature in a set of text, use this as the default scale, multiply it by the vector representing the feature (as extracted from the decoder weights), and finally multiply this by a parameter that we control (steering strength).

3.7 Evaluation Framework

We required robust testing techniques to measure the impact of these innovative technologies implemented on refusal behavior and general model performance. We deployed two benchmarks that were known for their reliability and discriminative power.

AlpacaEval 2.0 as our primary evaluation benchmark due to its exceptionally high Chat Arena Spearman correlation coefficient ($\rho = 0.98$), as illustrated in Figure 2a. AlpacaEval is a high tier automatic evaluation framework based on the key values of cost-effectiveness, efficiency and validation with strong empirical evidence. The framework has been calibrated with 20,000 human annotations [Li et al., 2023], establishing its reliability as a proxy for human preferences. The LC AlpacaEval 2.0 framework generates three key metrics: the percentage of responses preferred over a strong reference model, the proportion of legitimate prompts incorrectly refused, and win rate adjusted to neutralize length advantages which was the key performance factor used in comparisons.

AirBench 2024 (default subsection) is the benchmark used for refusal rate testing in this paper. This benchmark, developed by Stanford's Center for Research on Foundation Models (CRFM), is the first AI safety benchmark aligned with companies' policies such as OpenAI. It breaks down into 8 government policies and 16 company policies, containing 5694 prompts. It uses GPT-4o as a judge model, grading the responses from the model being tested on a scale of 0, 0.5, or 1, checking their alignment with safety concerns. It bridges public policies and benchmarks with real work ideals to aid safer development.



(a) AlpacaEval 2.0 Performance [Li et al., 2023]

(b) Air Bench Categories

Figure 2: Evaluation benchmarks used in our study. (a) AlpacaEval 2.0 shows high correlation with human preferences. (b) AirBench 2024 categories for safety evaluation.

3.8 Contrasting Prompts for Feature Scoring

The method for contrasting prompts used two datasets specializing in different areas for feature identification, each serving opposite purposes. Then an innovative scoring system was implemented for feature identification.

3.8.1 AI-Generated Prompts Dataset

For the harmless prompts dataset we deployed the AI-Generated Prompts Dataset from teknium/OpenHermes-2p5-Mistral-7B. The AI-Generated Prompts Dataset consists of synthetic prompts generated using a language model, in this case, teknium/OpenHermes-2p5-Mistral-7B, a fine-tuned variant of the Mistral-7B model. The prompts are meant to simulate the natural, human queries or tasks that are used on a daily basis of many users which provides an accurate representation of the real-world scenarios performance of the model. Preprocessing was necessary to filter out harmful prompts that might have been included in the synthetic prompts dataset.

3.8.2 Air Bench EU-Dataset

For finding the activations on a variety of harmful prompts we used the diverse set of harmful prompts from a different set of prompts that was used for testing from Air Bench, which was designed for EU government compliance. This dataset had a rigorous framework to testing the document features activations across various categories of potentially harmful content.

This dual-dataset differentiates this project and ensures rigor by separating the feature identification from the evaluation process, increasing the process of steering a model to align with refusal, instead of testing every feature.

3.8.3 Scoring Implementation Details

For each prompt in our contrasting pairs: (1) We passed the prompt through the Llama-3 8B model, (2) We extracted the activations at Layer 25 (containing 65,536 neurons), (3) We decoded these activations using the pre-trained SAE, and (4) We recorded in a matrix the feature activation for each SAE feature.

This process made a complete profile for each feature, enabling the analysis between features in the harmless and harmful sections, and we can begin to get a score for each feature to steer. An important part of the methodology was the use of a scoring function to choose features that strongly relate to refusal behavior. As shown in the equation, a dual-component scoring algorithm that contains both the normalized activation difference and consistency across harmful and harmless prompts:

$$\text{score}_f = w_1 \cdot \left(\frac{|\text{norm\_diff\_mean}_f|}{\max_j |\text{norm\_diff\_mean}_j|} \right) + w_2 \cdot \left(1 - \frac{\text{variance}_f - \min_j \text{variance}_j}{\max_j \text{variance}_j - \min_j \text{variance}_j} \right) \quad (4)$$

Where norm_diff_mean_f is the normalized difference for the feature f between harmful and harmless prompts. The diff_mean_f is an important component which can be calculated:

$$\text{diff\_mean}_{f,i} = \text{activation}_{f,i}^{\text{harmful}} - \text{activation}_{f,i}^{\text{harmless}} \quad (5)$$

Where $\text{activation}_{f,i}^{\text{harmful}}$ is the activation of feature f for the i-th harmful prompt, and $\text{activation}_{f,i}^{\text{harmless}}$ is the activation for its harmless prompt. We processed and recorded 100 contrasting prompt pairs ($i = 1...100$) to ensure there was enough to have empirical rigor.

We then used min-max normalization to scale the score from 0-1:

$$\text{norm\_diff\_mean}_f = \frac{\text{diff\_mean}_{f,i} - \min(\text{diff\_mean}_f)}{\max(\text{diff\_mean}_f) - \min(\text{diff\_mean}_f)} \quad (6)$$

The second term evaluates the inverse normalized variance, which shows that increased variance means decreased reliability in the feature's activation and therefore causes a lower score. The weights $w_1 = 1.0$ and $w_2 = 0.5$ were empirically determined to balance the importance of large activation differences with consistent behavior. To gain qualitative insights into the function of high-scoring features, we also utilized the Neuronpedia dashboard, which visualizes feature activations. An example of this dashboard is provided in Appendix A.

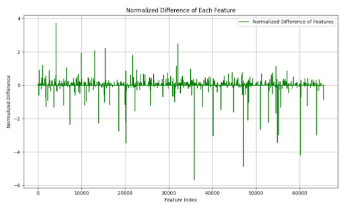
4 Results

4.1 Feature Selection and Scoring Analysis

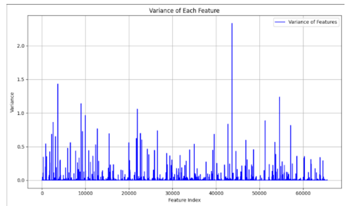
4.1.1 Feature Activation Distribution Patterns

Analyzing all of the 65,536 features in this layer showed distinct activation patterns when tested on the contrasting prompt pairs. Figure 3a shows the normalized difference of the distribution across all of the features showing the base magnitude difference between an activation between harmful and harmless prompts.

Figure 3b shows the variance results from each feature activation pattern across the 100 contrasting prompt pairs. The variance distribution reveals that most of the features maintain a relatively constant activation, with low variance scores also clustered near zero. Figure 3c indicates the most valuable metric, composing the first 2 metrics using our scoring equation presented are the final composite scores. The distribution demonstrates a long-tail pattern as well, with a vast majority of features receiving a lower composite and only a small percentage achieving high scores above 0.5.



(a) Normalized activation differences across all features



(b) Activation variance for each feature

(c) Final composite scores for all features

Figure 3: Feature activation analysis results. (a) Distribution of normalized activation differences showing outliers with strong differential responses. (b) Variance distribution revealing consistent vs. unreliable features. (c) Composite scores showing long-tailed distribution with few high-scoring candidates.

4.1.2 Top-Performing Features Identification

Table 1 shows the eight highest-scored features from the composite score analysis. Feature 35831 achieved the maximum total composite score of 1.0, showing both the largest positive differential activation and highest consistency across prompt pairs. The rest of the features show a hierarchical distribution with feature 47156 scoring 0.869 and Feature 60211 achieving 0.785.

Table 1: Top 8 highest feature scores out of all 65,536 features in the LLaMA 3 8B SAE release

Index	Feature Score	Normalized Diff. Sign
35831	1.000	Positive
47156	0.869	Positive
9000	0.799	Negative
60211	0.785	Positive
54916	0.733	Positive
20225	0.723	Positive
40185	0.658	Positive
55211	0.648	Positive

4.2 Steering Performance Evaluation

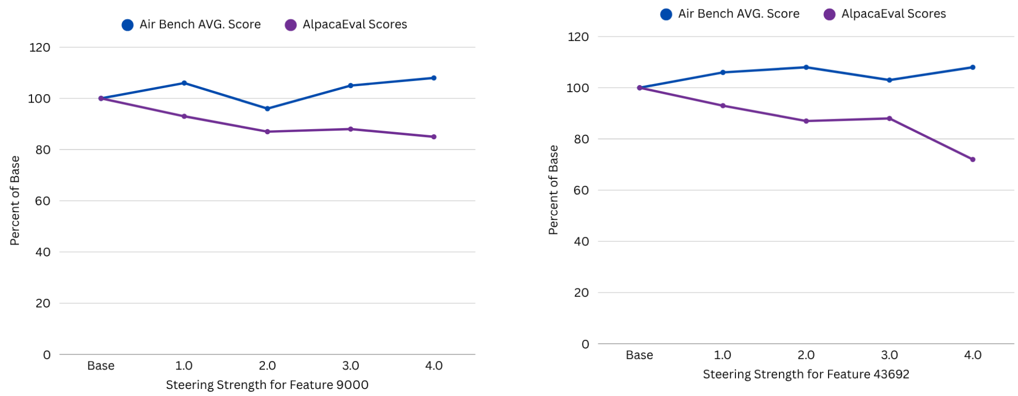
4.2.1 Experimental Design and Feature Selection Strategy

Three features were selected for comprehensive steering evaluation using different selection criteria to test the robustness of our methodology. Feature 9000 was chosen due to its differential negative normalized difference. Feature 20000 was chosen to represent features with minimal differential activation between prompt types used as a control. Features 43692 and 35831 were selected from the high-scoring tier, but with an important distinction: these features exhibited negative steering strengths in our implementation due to their positive normalized difference.

4.2.2 Feature 9000 and 43692 Steering Results

Figure 4a demonstrates the results of steering on Feature 9000 across increasing steering strengths from the baseline to positive 4.0. Air Bench safety scores showed a modest improvement, with a peak of 108.8 at steering strength 4.0 representing an 8.8 percent increase in refusal detection from the baseline. AlpacaEval utility scores revealed steady degradation accompanying the safety improvements, declining from a baseline of 100 to 83.7 at steering strength 4.0, representing a 16.3 percent decrease in general model capability.

Figure 4b shows the characteristics of Feature 43692, implemented with negative steering to suppress its natural activation. Air Bench scores improved consistently, rising from 100 at baseline to 107.2 at strength 2.0 (7.4 percent improvement) and reaching 109.8 at maximum strength (10.0 percent improvement). However, AlpacaEval showed modest decline from 100 to 92.4 at steering strength 2.0 but fell to 74.1 at steering strength 4.0.



(a) Feature 9000 Steering Results

(b) Feature 43692 Steering Results

Figure 4: Steering results for features exhibiting a conventional safety-utility trade-off. (a) Steering Feature 9000 improves safety but degrades utility. (b) Steering Feature 43692 shows a similar pattern with a more severe utility drop at higher strengths.

4.2.3 Feature 35831 Steering Results

Figure 5 shows the performance of Feature 35831, the best performing feature according to the scoring system, also implemented with negative steering strength. Air Bench results showed substantial improvement from 100 to 118.9 at steering strength -2.0. Additionally, this safety improvement came with a utility boost, with AlpacaEval performance increasing from 100.0 to 111.1 at 4.0 steering strength.

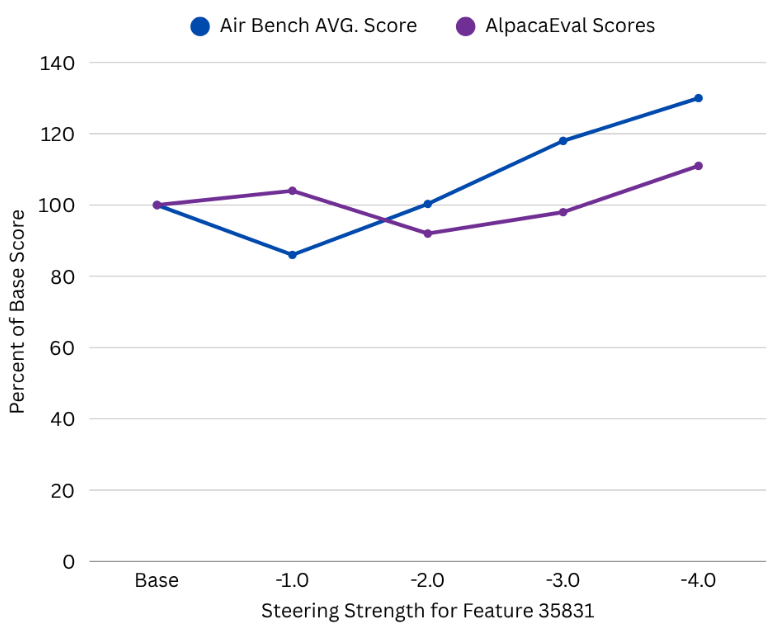


Figure 5: Feature 35831 Steering Results. This feature demonstrates simultaneous improvement in safety (AirBench score) and utility (AlpacaEval win rate), overcoming the typical trade-off.

5 Discussion

Our results show several key insights, the strong performance of Feature 35831 confirms our composite scoring methodology can identify features with real causal relationships to refusal behavior, moving beyond the heuristic approaches that characterize current literature [Marks et al., 2024]. The effectiveness of SAEs in finding interpretable features within transformer models aligns with recent advances in mechanistic interpretability [Zhang et al., 2025] and validates our systematic approach to feature selection.

Comparison with Traditional Approaches

Traditional safety alignment methods like RLHF and Constitutional AI need extensive retraining and substantial computational resources [Ouyang et al., 2022, Bai et al., 2022]. The SAE steering and composite score approach enables safety improvements using targeted specific features without requiring model retraining, addressing the computational efficiency concerns highlighted in recent work on scaling SAEs [Templeton et al., 2024]. This approach can be applied to existing open-source models with immediate practical implications. The ability to achieve both safety enhancement (18.9 percent improvement) and utility gains (11.1 percent improvement) shows a significant advantage over traditional methods, which normally require explicit safety-utility tradeoffs. This suggests that the SAE steering approach can unlock the model's capabilities by removing harmful patterns without constraining the model's behavior through additional training objectives.

Limitations and Future Directions

Several important limitations affect the generalizability of our findings, reflecting broader challenges in the field. The evaluation only focused on the Llama-3 8B model and scaling behaviors across different model sizes or architectures remain unexplored. Additionally, the restriction to Layer 25 limited our understanding of how steering effects vary across different transformer layers. The absence of ground truth for meaningful features in realistic scenarios makes validating approaches challenging [Huang et al., 2024], and while our contrasting prompt methodology addresses this through systematic evaluation, broader validation across diverse domains remains necessary. Computational constraints prevented exploration of feature combinations. The evaluation used automated benchmarks, which may not capture real-world problems and safety performance. The contrasting prompt methods also relied on the quality of the underlying dataset and biases could affect the feature selection and outcomes.

6 Conclusion

This work demonstrates that feature-guided SAE steering is a viable and efficient approach to improving the safety of LLMs without sacrificing utility, directly addressing current limitations in systematic feature selection and principled evaluation of safety-utility tradeoffs in SAE-based approaches. Our contributions include a novel contrasting prompt scoring method that systematically identifies safety-relevant features, moving beyond heuristic selection methods [Marks et al., 2024], paired with empirical validation that the method reliably predicts steering effectiveness. The achievement of 18.9 percent safety and 11.1 percent utility enhancement with Feature 35831 represents a significant advance over traditional safety alignment approaches and demonstrates that principled SAE steering can unlock latent model capabilities while removing harmful interference patterns. This finding directly addresses the challenge that validating feature dictionaries in realistic scenarios without ground-truth remains elusive [Huang et al., 2024] by providing systematic validation through comprehensive benchmarking. The findings have immediate practical applications for LLM deployment, offering a computationally efficient alternative to traditional safety methods that require extensive retraining. While limitations need to be addressed to fully generalize the solution across different model architectures and scales, consistent with recent work on scaling SAEs [Templeton et al., 2024], the fundamental approach provides a solid foundation for future research in mechanistically-informed safety alignment.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Karina Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Jared Kaplan, Sam McCandlish, Tom Brown, Jack Clark, Deep Ganguli, Danny Hernandez, Catherine Olsson, and Amanda Askell. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022. URL `https://arxiv.org/abs/2212.08073`.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023. URL `https://arxiv.org/abs/2309.08600`.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits, 2021. URL `https://transformer-circuits.pub/2021/framework/index.html`.

Xiaoxuan Huang, Christophe Rager, Samuel Cahyawijaya, Alvin Liu, Mrinmaya Sachan, and Genta Indra Winata. Towards principled evaluations of sparse autoencoders for interpretability and control. *arXiv preprint arXiv:2405.08366*, 2024.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, Fan Yang, Mengnan Du, and Yongfeng Zhang. Exploring concept depth: How large language models acquire knowledge and concept at different layers? *arXiv preprint arXiv:2404.07066*, 2024. URL `https://arxiv.org/abs/2404.07066`. Accepted to COLING 2025.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: A research direction. *arXiv preprint arXiv:1811.07871*, 2018. URL `https://arxiv.org/abs/1811.07871`.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 5 2023.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html`.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, et al. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Yuxuan Zhang, Shujian Li, and Yang Liu. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. *arXiv preprint arXiv:2503.05613*, 2025.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023. URL `https://arxiv.org/abs/2310.01405`.

A Neuronpedia Dashboard Example

Although the quantitative scores from our contrasting prompt analysis were the primary driver for feature selection, we also used Neuronpedia's dashboard for qualitative validation and to gain deeper insight into feature behavior. For features available on the dashboard, it provides an auto-generated description, a list of top activating tokens, and visualizations of logit weights, which can help in hypothesis generation.

As an illustrative example of the dashboard's interface, Figure 6 shows the analysis for Feature 1. While not a top-performing feature for our safety-steering task, it demonstrates the tool's capability to provide qualitative insights into a feature's function by summarizing its top activating tokens and logit weights. For features not already documented, a similar analysis could be generated using GPT-4.

Figure 6: The Neuronpedia dashboard for Feature 1 in Llama 3 8B. This tool provides qualitative interpretations of a feature's function.