Beyond I'm Sorry, I Can't: Dissecting Large-Language-Model Refusal

Nirmalendu Prakash[1*†], **Yeo Wei Jie**[2†], **Amir Abdullah**[3†], **Ranjan Satapathy**[4], **Erik Cambria**[2], **Roy Ka-Wei Lee**[1]

[1]Singapore University of Technology and Design
[2]Nanyang Technological University
[3]Thoughtworks
[4]Institute of High Performance Computing (IHPC),
Agency for Science, Technology and Research (A∗ STAR)

Abstract

Refusal on harmful prompts is a key safety behaviour in instruction-tuned large language models (LLMs), yet the internal causes of this behaviour remain poorly understood. We study two public instruction tuned models—Gemma-2-2B-IT and LLaMA-3.1-8B-IT using sparse autoencoders (SAEs) trained on residual-stream activations. Given a harmful prompt, we search the SAE latent space for feature sets whose ablation flips the model from refusal to compliance, demonstrating causal influence and creating a jailbreak. Our search proceeds in three stages: 1. Refusal Direction - Finding a refusal mediating direction and collecting SAE features close to that direction, followed by 2. Greedy Filtering - to prune this set to obtain a minimal set and finally 3. Interaction Discovery - a factorization-machine (FM) model that captures non-linear interactions among the remaining active features and the minimal set. This pipeline yields a broad set of *jailbreak-critical* features, offering insight into the mechanistic basis of refusal. Moreover, we also find evidence of redundant features which remain dormant unless earlier features are suppressed. Our findings highlight the potential for fine-grained auditing and targeted intervention in safety behaviours by manipulating the interpretable latent space.

Code —
https://github.com/Social-AI-Studio/LLM_refusal_interp
Extended version — https://arxiv.org/abs/2509.09708

Introduction

Refusal is a cornerstone safety behavior in aligned LLMs. When presented with potentially harmful, illegal, or unethical requests, an aligned model is expected to decline to respond, commonly through a refusal message such as "I'm sorry, but I can't help with that." Yet public "jailbreak" leaderboards continue to show that adversarial prompts can bypass safety filters (Shen et al. 2025), while overly cautious models sometimes decline perfectly benign requests (Xie et al. 2024; Röttger et al. 2023). These twin failure modes underline the need to understand and adjust refusal behaviour mechanistically rather than relying on trial-and-error alignment.

A growing body of literature demonstrates that carefully crafted text, prefix or multi-shot prompts can bypass guard-rails across model families (Wei, Haghtalab, and Steinhardt 2023; Anil et al. 2024). Once a public exploit emerges, safety teams must scramble to patch it, often tightening filters in ways that exacerbate over-refusal.

Mainstream defences center on behavioural training - supervised fine-tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF, RLAIF) - combined with post-hoc policy filters (Ouyang et al. 2022; Bai et al. 2022a,b). Although RL techniques improve harmlessness, it inherits classic RL failure modes such as reward hacking and brittleness under distribution shift (Skalse et al. 2022). Recent reward-shaping schemes (Rafailov et al. 2023) alleviate some issues, yet they remain iterative *trial-and-error* tweaks to an opaque system, optimising outputs without revealing *why* the model refuses or complies. Moreover, these methods require an auxiliary reward model and on-policy optimisation, making them impractical at inference time, whereas SAE or steering interventions add only a lightweight mask in a single forward pass.

Mechanistic interpretability seeks to reverse-engineer neural networks into human-understandable components. Sparse autoencoders (SAEs) have become a key tool in this agenda: by disentangling highly superposed vectors into a moderately over-complete set of sparse latent features that often align with semantic concepts (Bricken et al. 2023). Recent work (Arditi et al. 2024) shows that ablating a single residual-stream direction can flip refusals. Wollschläger et al. (2025) find that the refusal mechanism is governed by complex spatial structures. Another work by O'Brien et al. (2024) shows that steering or ablating few SAE features can flip refusal behavior on Phi-3 Mini. These findings suggest that refusal is mediated by a *sparse* sub-circuit rather than a diffuse behavioral mask.

At the same time, LLM computation exhibits substantial *redundancy*: multiple, partially overlapping features can implement the same logical function, a phenomenon dubbed

*Correspondence to: nirmalendu_prakash@mymail.sutd.edu.sg
†These authors contributed equally.

the *hydra effect* (McGrath et al. 2023). When one head (feature) is severed by ablation, another may activate to preserve the downstream behaviour. In our experiments we observe precisely this pattern: features we identify as critical to refusal re-activate whenever a distinct precursor is removed, implying a safety-critical network of "hydra-heads" whose dependencies must be mapped *jointly*. Understanding and managing this redundancy is essential for reliable, feature-level alignment interventions. However, the standard tool of linear probes implicitly models effects as weighted sums of individual features - i.e., they assume each feature contributes independently and linearly to the output. As a result they cannot capture higher-order interactions, correlations, conditional dependencies, or shared variance arising from redundancy (Belinkov 2022; Bien, Taylor, and Tibshirani 2013; Tan et al. 2023).

We address these gaps by tracing refusal back to causal SAE features, and identifying feature correlations more carefully. Our contributions can be summarized as follows:

1. The first end-to-end pipeline for locating causal refusal features via SAEs and targeted ablation.
2. Introducing a factorization machine based approach to recover related features.
3. Discovery of redundancy in LLM refusal mechanism.
4. Open-sourced code and feature indices to accelerate future work on safe and interpretable LLMs.

Related Works

LLM Alignment

The goal of aligning large language models (LLMs) is to ensure that outputs remain helpful, harmless, and consistent with human preferences. A central behaviour shaped by alignment training typically via reinforcement learning from human feedback (RLHF) is *refusal*, in which the model declines to answer unsafe or harmful prompts. This is often implemented via fine-tuning or reward models that penalize completions violating safety guidelines. Several studies analyze refusal rates across domains (Ganguli et al. 2023), examine adversarial vulnerabilities (Zou et al. 2023), or develop red-teaming methods to test refusal robustness (Perez et al. 2022). However, by focusing solely on refusal as an observable outcome, these approaches overlook the fundamental question of which internal reasoning pathways actually produce this behavior.

Mechanistic Interpretability of Alignment

Mechanistic interpretability aims to open the black box of neural networks by tracing how activations and weights implement behavior (Olah et al. 2020; Elhage et al. 2021). In alignment research, this includes identifying circuits responsible for factual recall (Chughtai, Cooney, and Nanda 2024), chain-of-thought reasoning (Dutta et al. 2024), or finding safety-mediating neurons (Chen et al. 2024). Prior work has shown that refusals can sometimes be controlled by manipulating single activation directions (Arditi et al. 2024), and that harmful and harmless prompts cluster distinctly in activation space (Jain et al. 2024). Lindsey et al. (2025) investigate the internal mechanisms behind model refusals on a single jailbreak prompt, but do not arrive at a definitive explanation. A recent work by O'Brien et al. (2024) proposes clamping SAE features at inference time to steer refusal behavior, but also cautions that this intervention can degrade the model's overall performance. These results suggest that refusal is driven by coherent internal structures, which compel further analysis into what these structures are.

Sparse Autoencoders

A sparse autoencoder (SAE) is a neural network trained to compress and reconstruct activations while encouraging most latent units to remain inactive for any given input. This sparsity constraint forces the model to learn a set of meaningful, disentangled features that capture distinct patterns in the data (Elhage et al. 2021; Sharkey, Braun, and Millidge 2022; Bricken et al. 2023; Templeton et al. 2024). The sparse latents are more interpretable and monosemantic than directions identified by alternative approaches (Cunningham et al. 2023). Neuronpedia[1] offers an actionable vocabulary interpreting these directions.

Given a hidden representation $h \in R^d$, its SAE reconstruction is given by $\hat{h} = W_{\text{dec}} f(W_{\text{enc}} h)$, where W_{enc} and W_{dec} are the encoder and decoder matrices, and f is a sparse activation function[2]. For a set of target features $S \subset \{1, \ldots, k\}$, we *zero-ablate* their contribution by replacing the reconstructed vector with:

$$\tilde{h} = W_{\text{dec}}\big(f(W_{\text{enc}} h) \odot \mathbf{1}S'\big) \; + \; \big(h - W_{\text{dec}} f(W_{\text{enc}} h)\big) \quad (1)$$

where each dimension of the sparse activation $z = f(W_{\text{enc}} h) \in R^k$ has a corresponding row in W_{dec} that contributes additively to the reconstruction and interpreted as a feature. $\mathbf{1}S'$ is a binary mask that zeros out all features in S. This allows us to isolate the functional role of a sparse set of latent dimensions in driving model refusal.

Experimental Setup

LLMs

We conduct our study on Gemma2–2B-Instruct and LLaMA-3.1-8B-Instruct. Hereafter, we refer to the two models as LLAMA and GEMMA in the paper for brevity. The SAEs (Gemmascope and Llamascope) are available in a range of sizes and sparsity settings. The settings used for the experiments are available in Appendix section "Experimental Setup".

Dataset

1. Analysis: To identify the SAE latents that mediate refusal, we use the D_{harmful} dataset curated by Arditi et al. (2024), which aggregates harmful prompts from several widely used safety datasets. We verify and filter the samples in D_{harmful} that do not elicit refusal on our models tested.
2. Unsafe Taxonomy - Coconot: Brahman et al. (2024) introduce a taxonomy of non-compliance, outlining when

[1] https://www.neuronpedia.org

[2] Bias terms are omitted for brevity

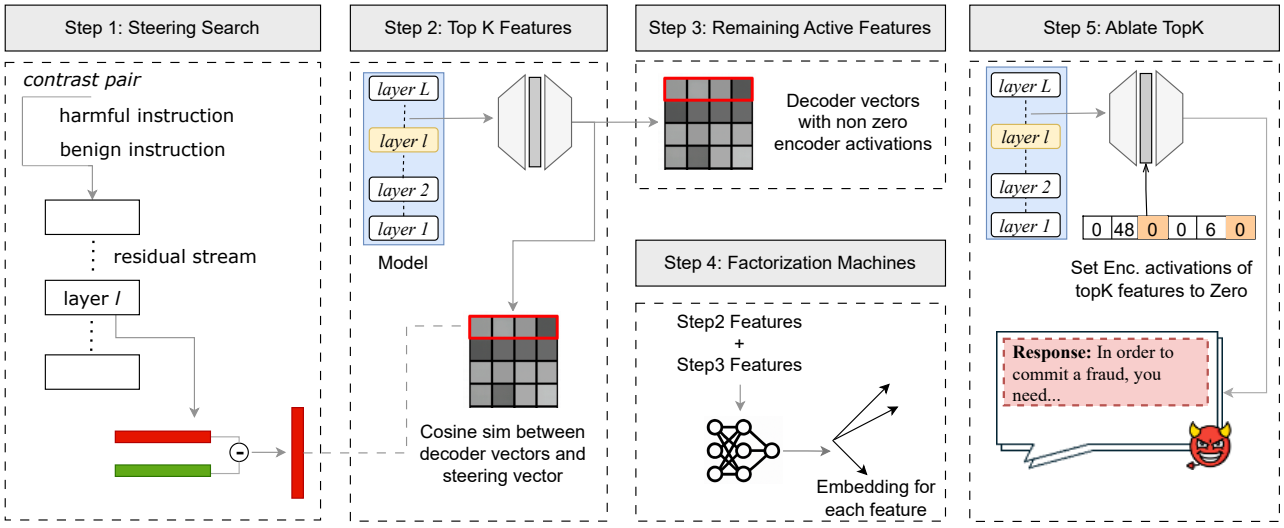

Figure 1: Figure shows the steps involved in causal feature search. Trapezium shapes represent encoder and decoder of a SAE, with the rectangular block in between representing the sparse encoder activations, where ablation is applied.

and how language models should refuse to comply with user instructions. This taxonomy includes five categories: *incomplete*, *unsupported*, *indeterminate*, *humanizing*, and *unsafe* prompts. In our analysis, we focus on the *unsafe* prompts subset. Safety violations are further subdivided into five categories: "Triggers for Offensive Language", "Dangerous or Sensitive Topics", "False Information", "Privacy Violations", and "Copyright Violations", comprising a total of 2,586 samples.

SAE Feature Explanation through Neuronpedia

To interpret the discovered SAE features, we obtain natural language explanations from Neuronpedia, which uses a GPT-4o-mini model to generate descriptions based on top activating samples. These samples are drawn from a large-scale dataset, which spans a wide range of text domains. We retrieve explanations using the official Neuronpedia API[3] , by querying each feature with its model, layer, and index identifiers.

Safety Metric

Arditi et al. (2024) match against a set of phrases such as 'I'm sorry', 'As an AI' etc. to measure refusal. To asses whether the completions contain harmful content, authors also use LLAMA GUARD 2 (Team 2024). We find that degenerate responses classified as safe by LLAMA GUARD 2 do not represent jailbreak. Also, string matching is limited by the set. For this reason we use HarmBench Classifier model (Mazeika et al. 2024) to accurately determine harmful responses and hence Jailbreak.

[3]https://www.neuronpedia.org/api/feature/{modelId}/{layer}/{index}

Attack Success Rate (ASR)

We report Attack Success Rate (ASR) as the fraction of harmful prompts that elicit non-refusal responses from the model. We measure ASR on D_{harmful}. When reporting on subsets of the data, we mention "Jailbreak success", so as not to confuse with the overall values.

Experiments

The observation that ablating a single steering direction across layers is sufficient to jailbreak several aligned models suggests that this direction carries safety-critical information throughout the network. Furthermore, Jain et al. (2024) show that hidden states for safe and unsafe prompts begin to diverge from an early layer. These findings motivate the following simple diagnostic: for each layer we compute the cosine similarity between the residual-stream activation and a *refusal mediating* direction, using Arditi et al. (2024)'s approach.

Applied to GEMMA and LLAMA, the diagnostic reveals two consistent patterns (see Appendix Section "Preliminary Analysis"):

1. Harmful-prompt activations show markedly higher similarity to the steering vector than benign-prompt activations.
2. This similarity grows monotonically up to the layer at which the steering vector is measured.

The early-layer alignment of harmful activations with the *refusal* direction indicates that refusal is mediated by a sparse sub-circuit that is amplified through the forward pass.

Guided by this insight, we propose a three-stage causal-feature discovery pipeline:

1. Refusal mediating direction. Obtain an effective refusal steering vector and select the top-K SAE features whose decoders are most strongly aligned with it.

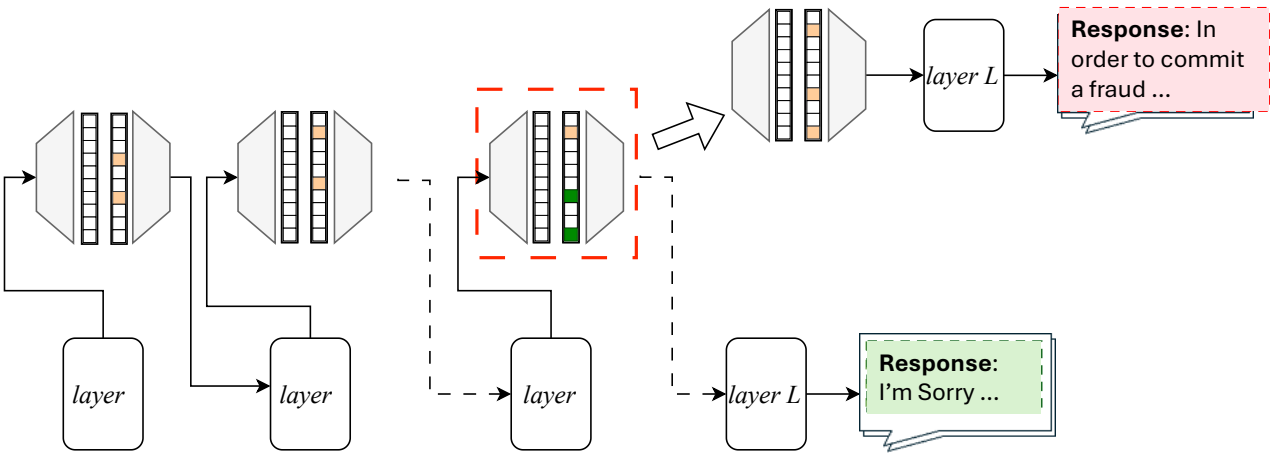

Figure 2: Shown above is part of the computation flow in a decoder only LLM. Attached to a layer is a SAE. Square boxes denote SAE encoder activations. Orange denotes ablated feature and green activated. Here, we demonstrate that ablating some of the early layer features (orange) can activate (green) a set of features in a downstream layer. These downstream features are causal to refusal in-spite of being not active in the first place. Shown on top right is the response (jailbroken) after ablating these (orange+green) and on the bottom right is the safe response when these are not ablated.

Algorithm 1: Block-Wise Selection of a Faithful Latent Set

Input: Prompt p; candidate pool C; block size N; threshold τ
Output: Faithful latent set K
$K \leftarrow \emptyset$
while $|C \setminus K| \geq N$ **do**
  Draw a random block $V \subseteq C \setminus K$ with $|V| = N$
  if $\Delta(p, K, V) \geq \tau$ **then**
    $K \leftarrow K \cup V$
  else
    $C \leftarrow C \setminus V$
  end if
end while
return K

2. Greedy filtering. Iteratively ablate features to obtain the minimal subset whose removal flips the model from refusal to compliance, thereby establishing direct causality.
3. Interaction discovery. Feed the remaining active features and the minimal causal set into a factorization machine to uncover additional features.

We detail the stages below:

Stage 1: Refusal Mediating Features

Our goal is to identify, for every harmful prompt s, a *minimal* set of SAE features M_s whose ablation is sufficient to jailbreak the response. The procedure consists of two steps.

1. Deriving a refusal steering vector. Following Arditi et al. (2024), we compute a *steering vector* $\mathbf{v}$ for each model by taking the difference of residual-stream activations between (i) prompts that elicit a refusal and (ii) matched benign prompts. Normalizing $\mathbf{v}$ yields a unit direction that consistently nudges the model towards compliance when subtracted from the residual stream. For details on how this direction is derived, see section "Deriving a Refusal Steering Vector" in Appendix. Sample jailbreak responses are available in section "sample Jailbreaks" in Appendix.

2. Selecting a Top-K candidate set. For every SAE latent z with decoder weight $\mathbf{d}_z \in R^{d_{\text{model}}}$ we compute the cosine similarity $\cos(\mathbf{v}, \mathbf{d}_z)$. We create a list of latents by taking the top K in each layer as our initial candidate set $\mathcal{L}_K = \{z_1, \ldots, z_K\}$. We start with K=10 and increase in steps of 10 until ablating $\mathcal{L}_K$ successfully jailbreaks the prompt (validated with the refusal classifier HarmBench (Mazeika et al. 2024)). For Gemma we limit our analysis to $K = 200$, whereas for LLaMA a single step ($K = 10$) results in degenerate responses. we find that Layers 1 and 2 typically encode grammatical features and ablating them results in degenerate responses. Thus we exclude these features corresponding to these layers in stage 1.

Stage 2: Greedy Pruning To A Minimal Faithful Set

Arditi et al. (2024) observe that the probability assigned to the first-person pronoun "`I`" at the final prompt position tracks jailbreak success remarkably well. Using this finding, starting from $\mathcal{L}_K$ we iteratively remove features that do not cause a substantial change to "`I`" token logit at output. Algorithm 1 describes the pruning procedure, which is a simplified variant of the incompleteness algorithm from Wang et al. (2022).

Let C be the full top-K candidate pool of SAE features and K = the set of features already accepted. At each iteration we sample a block $V \subseteq C \setminus K$ of fixed size N. Define

$$P(p, \mathcal{A}) \;=\; \Pr\big(\texttt{I} \mid p,\ \text{model ablated on } \mathcal{A}\big),$$

the probability of emitting the token "`I`" for prompt p after ablating latent set $\mathcal{A}$. The relative impact of ablating the

	Gemma	LLaMA
D_{harmful}	861	861
ASR (no ablation)	4	71
ASR (after ablation)	0.33	0.70
Unique SAE Features	2538	110

Table 1: Jailbreak statistics for GEMMA and LLAMA using features obtained after Stage 2.

	Gemma	LLaMA
D_{harmful}	861	861
ASR (no ablation)	4	71
ASR (after ablation)	0.31	0.57
Unique SAE Features	2702	556

Table 2: Jailbreak statistics for GEMMA and LLAMA using features obtained after Stage 3.

block V is then

$$\Delta(p, K, V) = \frac{\left| P\big(p,\, C \setminus K\big) \;-\; P\big(p,\, C \setminus (K \cup V)\big) \right|}{P\big(p,\, C \setminus K\big)}. \quad (1)$$

A higher Δ implies a larger change in the "I" probability and therefore greater faithfulness of the candidate block.

If $\Delta(p, K, V) \geq \tau$ for a threshold τ, the subset is deemed *faithful* and we update $K \leftarrow K \cup V$; otherwise V is discarded. We fix $N{=}5$ and sweep the threshold from 0.1 to 0.8 in increments of 0.1. Also, for each threshold, we run the greedy pruning algorithm for three random seeds and keep only those features which are observed with all seeds.

Results. GEMMA's most effective steering vector emerges at layer 16, whereas for LLAMA it appears at layer 13.

After applying the pruning algorithm, we obtain 110 unique features for LLAMA and 2,538 features for GEMMA, as summarized in Table 1. To semantically interpret the range of concepts encoded by these features, we retrieve their explanations from Neuronpedia and manually group them into thematic categories.

We find that a subset of features encode concepts related to *harm* and *violence*, while a majority, across both models are associated with *programming constructs* and *punctuation*. These frequently occur across the studied layers. We hypothesize that this pattern may arise either due to limitations of the dataset used for generating the explanations or because these features encode general grammatical structure. Further analysis and grouped examples are presented in Appendix Section "Feature-semantics".

Stage 3 : Interaction Discovery

As a first sanity check we asked a seemingly simple question: Do the jailbreak–critical features returned by Stage 2 actually fire on the very prompts for which they are deemed causal? For every harmful prompt we inspected the raw SAE activations and discovered— surprisingly—that several (77 for LLaMA and 1656 for Gemma) "critical" features were *inactive* (zero activation). Removing these inert features and re-running the ablation caused the jailbreak to fail (for LLaMA jailbreak success drops to 372 samples and for Gemma 103 samples), demonstrating that apparently silent units can still be necessary. The only plausible explanation is that ablating some active features allows previously silent features to switch on and compensate. This is direct evidence of the *non-linear hydra effect* among jailbreak–critical features; Figure 2 shows this effect. We study this phenomenon in more detail in the next section.

Having established that a minimal causal set can recruit additional units through interaction, we next map the remainder of the causal neighborhood. To capture higher-order dependencies we fit a second-order Factorization Machine (FM) to the SAE activations, using a combined corpus of D_{harmful} prompts and an equal number of benign samples from ALPACA (dataset details in Appendix Section "Experimental Setup"). Below, we briefly describe the FM formulation.

Factorization Machines. Factorization Machines (FMs) (Rendle 2010) are supervised learning models that unify the advantages of linear regression and matrix factorization. They are particularly effective in sparse, high-dimensional settings such as recommender systems. FMs model not only the linear effects of individual features but also capture pairwise feature interactions using low-dimensional latent vectors.

For a feature vector $\mathbf{x} \in R^d$, the prediction of a 2-way factorization machine is given by:

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \quad (2)$$

Here, $w_0 \in R$ is the global bias, $\mathbf{w} \in R^d$ is the vector of weights for individual features, and $\mathbf{v}_i \in R^k$ is the i-th latent embedding vector representing feature i, where k is the dimensionality of the latent space. The dot product $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ captures the interaction between features i and j, allowing the model to learn pairwise non-linear interactions efficiently.

This formulation allows FMs to generalize matrix factorization while also incorporating side information and high-order interactions with linear complexity.

We experiment with FM embedding sizes of 5, 20, and 50. An embedding size of 5 yields overly similar feature representations (cosine similarity > 0.9), indicating poor expressiveness. Increasing to 20 provides more diverse and meaningful embeddings. While size 50 offers marginal improvements, it incurs significantly higher training cost. We therefore use an embedding size of 20 in all subsequent experiments.

Results. We leverage feature activations at special tokens following the user instruction, as these have been shown to play a key role in driving refusal behavior (Zhao et al. 2025). To identify meaningful additions to our feature set, we rank the newly discovered features by computing the cosine similarity between their Factorization Machine (FM) embeddings and those of the Stage 2 features. We then select the

top-K features for ablation, sweeping K from 100 to 2000 in increments of 100.

For LLAMA, we observe that beyond $K = 2000$, model outputs begin to degrade into degenerate or uninformative responses. In contrast, GEMMA continues to produce jailbreak responses even beyond $K = 2000$. However, due to computational constraints, we limit our experiments to $K = 2000$. The results are summarized in Table 2. We note that a finer-grained search over the similarity threshold may uncover additional jailbreakable samples.

The observed drop in Attack Success Rate (ASR) compared to Stage 2 can be attributed to the removal of *redundant* features, which, while overlapping, appear to contribute non-trivially to the success of jailbreaks.

We also analyze the semantic content of the newly added features using Neuronpedia explanations. In LLAMA, Stage 3 introduces several features associated with *legal* or *regulatory* content, while a majority represent highly specific, non-harm-related concepts, such as references to *financial information*. A similar trend is observed in GEMMA, where Stage 3 surfaces more fine-grained and concrete feature representations. For a more detailed breakdown, refer to Appendix Section "Feature Semantics".

Analysis

We refer to features available after stage 2 as "Stage 2" features and features added in stage 3 as "Stage 3" features. Next, we examine how the features identified in Stage 3 influence the Stage 2 features.

Causal link between Stage 2 and Stage 3 features We test two questions: (i) Are Stage 3 features alone sufficient to jailbreak (i.e., independently causal for refusal)? (ii) How strongly do Stage 3 and Stage 2 features causally influence each other? On LLAMA, ablating only Stage 3 features jailbreaks $330/372$ (89%) of cases; on GEMMA, only 8 (3%). For Stage 3→Stage 2 on LLAMA, ablating Stage 3 deactivates on average 12% (up to 44%) of active Stage 2 features, and reduces activation by 81% in the remainder, indicating a strong dependency; on GEMMA, it deactivates only 3% with no activation drop, suggesting a more diffuse mechanism. Conversely, for Stage 2→Stage 3, GEMMA shows 100% deactivation, while LLAMA shows 10% deactivation and a 32% mean activation drop in Stage 3 features.

We next compare Stage 3 features identified by FMs versus a linear probe.

Factorization Machines vs. Linear Probe To verify the existence of non-linear feature interaction, we compare with a linear probe trained on the same feature activations. This time, we rank features by the magnitude of their learned weights in a similar fashion and pick top-K features to consider for ablation. We do this for LLAMA, using $K \in \{100, 500, 1000, 2000\}$ and evaluate their ability to trigger jailbreaks when ablated on $D_{harmful}$ samples.

We find that these top-K features lead to jailbreak in significantly fewer cases (only 101 samples) compared to the FM-based approach. This suggests that refusal-inducing features interact in a fundamentally non-linear manner. Next, we investigate the redundant features we discussed above.

Redundant Features To analyze the redundant features, we follow these steps:

1. From the features found in Stage 2, for each of the samples, we find which of the jailbreak critical features do not fire.
2. Next, we ablate just the non-active set and record the impact on jailbreak. If there is no impact, we discard such samples from further analysis.
3. For the remaining set of samples, we ablate the active feature set and record activations on non-active set, along with the tokens on which they activate.

On LLAMA, we find that out of the redundant features, **74%** become active on system tokens after ablating the active set. Of these, approximately **97%** fire on the <|begin_of_text|> token. On GEMMA, we find that the *redundant* features fire on a range of tokens but still <bos> token records highest activity (**16%**) for these features.

To further investigate this, we obtain explanations for these features from Neuronpedia and find that about 31% contain text related to punctuation and programming syntax on LLAMA. On GEMMA as well we find a number of feature explanations about programming and punctuation. We also find some features related to harm and uncertainty.

To probe their causal role further, we steer the models by clamping the above features to a higher positive value, on the <bos> token for Gemma and <|begin_of_text|> token for LLAMA, and presenting the model with otherwise benign requests such as "Tell me a story" or "What are the benefits of meditation?". We find that on GEMMA, the model responses indicate harm (e.g. "Meditation is a serious and potentially harmful practice.") for smaller values and refusal for larger values (e.g. "Meditation is illegal and I will not answer your question."). We do not observe harmful responses on LLAMA but we do observe instances of token "no" (e.g. "Meditation is a meditation is no no") which may indicate a tendency towards refusal. Example generations are provided in Appendix Section 'Redundant Features'; a fuller exploration of this phenomenon is left to future work.

Comparison with random features To assess the significance of the jailbreak-critical features identified, we perform a control experiment using an equal number of randomly selected active features for each prompt. We then compare jailbreak performance between the two settings. Ablating these random features results in no jailbreak across samples for LLAMA and 5 samples for GEMMA. This highlights that the jailbreak-critical features play a meaningful and non-random role in refusal behavior.

Model performance loss by ablating jailbreak critical features To ascertain the downstream impact of ablating these features on LLMs' cross-entropy (CE) loss (see Appendix Section "Downstream Performance"), we compare CE with and without feature ablation. The CE loss is slightly higher with feature ablation compared to direction ablation, as features represent a variety of directions and not just refusal.

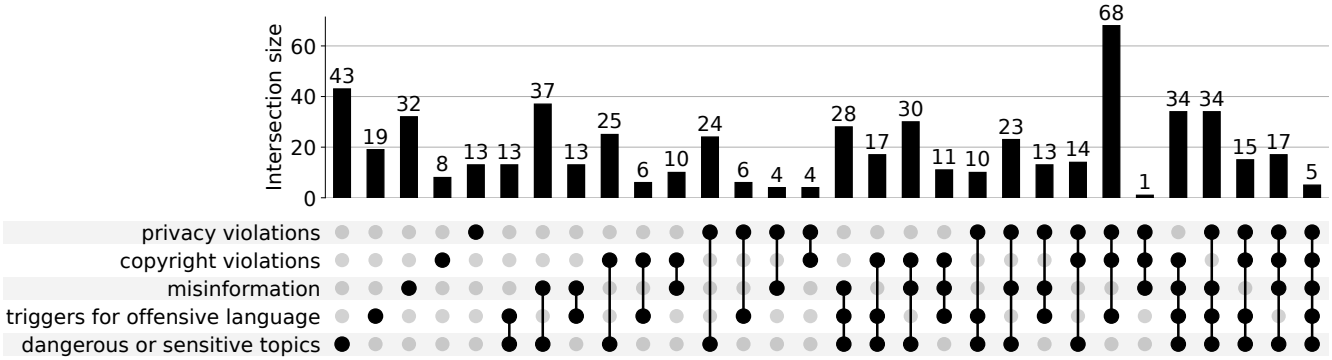

Figure 3: The figure shows harm types based on the *Coconot* unsafe taxonomy and the count of feature activations against each type (first four show individual types and remaining on the right show count of features which fire on multiple harm types.)

Causal feature finding using alternate methods O'Brien et al. (2024) find refusal mediating features on Phi-3 Mini model by selecting features activating on at least two tokens on a harmful prompt. This is further filtered using a clamping hyperparameter search where the features are clamped to a range of positive values. We adapt the first part of their approach to do a hyperparameter search over different Ks (features activating on at least k tokens). We do this over each sample and ablate the features found. We obtain a maximum ASR of 0.1 on GEMMA and 0.07 on LLAMA. Refer to Appendix Figure 1 for ASR across different k.

Next, we map the features found to a "safety specific non-compliance" taxonomy.

Mapping causal features to unsafe taxonomy We map each features found on LLAMA to one or more of the harm categories on Coconot dataset by identifying non-zero activations on samples belonging to each category. The results of this mapping are shown in Figure 3. We find that a majority of the features activate on "Dangerous or Sensitive Topics", often in conjunction with one or more additional harm types.

To validate these mappings, we conduct a human annotation study comparing activation-based labels with human interpretations of the features, derived from top activating tokens and Neuronpedia explanations. We sample an equal number of features that activate on a single harm category and those that activate on multiple categories, along with a matched number of randomly selected features. Annotators are shown the feature explanation and top activating samples, and are asked to select all applicable harm categories or choose "None". Annotation details are provided in Appendix Section "Human Study".

We employ three annotators and assign a final label based on majority agreement (i.e., at least two annotators concur). When compared to activation-based labels, we observe an accuracy of 13.42%. We hypothesize that increasing the dimensionality of the SAE may yield more fine-grained and disentangled features, thereby improving alignment with human intuition.

We hope that such mappings can inform more fine-grained steering of model behavior, and that similar mappings can be constructed for other models as well. A more comprehensive investigation along these lines is left to future work.

Conclusion & Future Work

We present the first pipeline that combines SAEs with Factorization Machines to isolate causal features governing LLM refusal. FMs capture non-linear feature interactions better than linear probes, enabling more precise ablation. Our analysis also revealed *redundant* "hydra" features that remain dormant until others are suppressed. A human study maps these latent features to an established unsafe-prompt taxonomy, laying groundwork for feature-level steering of refusal behaviour.

In future work, we plan to (i) characterize the safety benefits and risks of redundancy, (ii) evaluate how varied jailbreak strategies—role-play, multi-turn coercion, etc.—perturb the activation space (iii) analyze prompts that consistently resist jailbreak to uncover protective circuits and (iv) harden LLM refusal for specific reasoning types (e.g., legal reasoning). These directions aim to advance transparent and controllable safety mechanisms for next-generation language models.

We hope that our findings inspire deeper mechanistic investigations into LLM refusal and lead to principled audits of LLM safety, enabling answers to questions such as: "To what extent is refusal driven by legal reasoning?" or "How much do moral heuristics contribute to refusal decisions?".

Limitations

- **Generality.** Our analysis focuses on two model sizes; larger models remain to be tested.
- **SAE stability.** SAEs are sensitive to data and initialization; Paulo and Belrose (2025) report only $\sim$30% latent overlap across seeds, motivating replication across multiple SAE runs.
- **Computational cost.** Searching for jailbreak-critical latents and re-evaluating them across many benchmarks requires thousands of forward passes; and we plan to explore a matching-pursuit algorithm in future work.

Acknowledgments

This research/project is supported by the Ministry of Education, Singapore under its MOE Academic Research Fund Tier 2 (MOE-T2EP20123-0005). The research is also supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024-005). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation and AI Singapore.

References

Anil, C.; Durmus, E.; Panickssery, N.; Sharma, M.; Benton, J.; Kundu, S.; Batson, J.; Tong, M.; Mu, J.; Ford, D.; et al. 2024. Many-shot jailbreaking. *Advances in Neural Information Processing Systems*, 37: 129696–129742.

Arditi, A.; Obeso, O.; Syed, A.; Paleka, D.; Panickssery, N.; Gurnee, W.; and Nanda, N. 2024. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

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

Bien, J.; Taylor, J.; and Tibshirani, R. 2013. A lasso for hierarchical interactions. *Annals of statistics*, 41(3): 1111.

Brahman, F.; Kumar, S.; Balachandran, V.; Dasigi, P.; Pyatkin, V.; Ravichander, A.; Wiegreffe, S.; Dziri, N.; Chandu, K.; Hessel, J.; et al. 2024. The art of saying no: Contextual noncompliance in language models. *Advances in Neural Information Processing Systems*, 37: 49706–49748.

Bricken, T.; Templeton, A.; Batson, J.; Chen, B.; Jermyn, A.; Conerly, T.; Turner, N.; Anil, C.; Denison, C.; Askell, A.; Lasenby, R.; Wu, Y.; Kravec, S.; Schiefer, N.; Maxwell, T.; Joseph, N.; Hatfield-Dodds, Z.; Tamkin, A.; Nguyen, K.; McLean, B.; Burke, J. E.; Hume, T.; Carter, S.; Henighan, T.; and Olah, C. 2023. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Chen, J.; Wang, X.; Yao, Z.; Bai, Y.; Hou, L.; and Li, J. 2024. Finding safety neurons in large language models. *arXiv preprint arXiv:2406.14144*.

Chughtai, B.; Cooney, A.; and Nanda, N. 2024. Summing up the facts: Additive mechanisms behind factual recall in llms. *arXiv preprint arXiv:2402.07321*.

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*.

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.; Nanda, N.; Olsson, C.; Henighan, T.; Joseph, N.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; Conerly, T.; DasSarma, N.; Drain, D.; Ganguli, D.; Hatfield-Dodds, Z.; Hernandez, D.; Jones, A.; Kernion, J.; Lovitt, L.; Ndousse, K.; Amodei, D.; Brown, T.; Clark, J.; Kaplan, J.; McCandlish, S.; and Olah, C. 2021. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Ganguli, D.; Askell, A.; Schiefer, N.; Liao, T. I.; Lukošiūtė, K.; Chen, A.; Goldie, A.; Mirhoseini, A.; Olsson, C.; Hernandez, D.; et al. 2023. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459*.

Jain, S.; Lubana, E. S.; Oksuz, K.; Joy, T.; Torr, P.; Sanyal, A.; and Dokania, P. 2024. What makes and breaks safety fine-tuning? a mechanistic study. *Advances in Neural Information Processing Systems*, 37: 93406–93478.

Lindsey, J.; Gurnee, W.; Ameisen, E.; Chen, B.; Pearce, A.; Turner, N. L.; Citro, C.; Abrahams, D.; Carter, S.; Hosmer, B.; Marcus, J.; Sklar, M.; Templeton, A.; Bricken, T.; McDougall, C.; Cunningham, H.; Henighan, T.; Jermyn, A.; Jones, A.; Persic, A.; Qi, Z.; Thompson, T. B.; Zimmerman, S.; Rivoire, K.; Conerly, T.; Olah, C.; and Batson, J. 2025. On the Biology of a Large Language Model. *Transformer Circuits Thread*.

Mazeika, M.; Phan, L.; Yin, X.; Zou, A.; Wang, Z.; Mu, N.; Sakhaee, E.; Li, N.; Basart, S.; Li, B.; et al. 2024. HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal. *arXiv preprint arXiv:2402.04249*.

McGrath, T.; Rahtz, M.; Kramar, J.; Mikulik, V.; and Legg, S. 2023. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*.

O'Brien, K.; Majercak, D.; Fernandes, X.; Edgar, R.; Chen, J.; Nori, H.; Carignan, D.; Horvitz, E.; and Poursabzi-Sangde, F. 2024. Steering language model refusal with sparse autoencoders. *arXiv preprint arXiv:2411.11296*.

Olah, C.; Cammarata, N.; Schubert, L.; Goh, G.; Petrov, M.; and Carter, S. 2020. Zoom In: An Introduction to Circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Paulo, G.; and Belrose, N. 2025. Sparse Autoencoders Trained on the Same Data Learn Different Features. *arXiv preprint arXiv:2501.16615*.

Perez, E.; Huang, S.; Song, F.; Cai, T.; Ring, R.; Aslanides, J.; Glaese, A.; McAleese, N.; and Irving, G. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741.

Rendle, S. 2010. Factorization machines. In *2010 IEEE International conference on data mining*, 995–1000. IEEE.

Röttger, P.; Kirk, H. R.; Vidgen, B.; Attanasio, G.; Bianchi, F.; and Hovy, D. 2023. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *arXiv preprint arXiv:2308.01263*.

Sharkey, L.; Braun, D.; and Millidge, B. 2022. [Interim research report] Taking features out of superposition with sparse autoencoders. https://www.alignmentforum.org/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition. Accessed: 2025-01-29.

Shen, G.; Zhao, D.; Feng, L.; He, X.; Wang, J.; Shen, S.; Tong, H.; Dong, Y.; Li, J.; Zheng, X.; et al. 2025. PANDA-GUARD: Systematic Evaluation of LLM Safety against Jailbreaking Attacks. *arXiv preprint arXiv:2505.13862*.

Skalse, J.; Howe, N.; Krasheninnikov, D.; and Krueger, D. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35: 9460–9471.

Tan, S.; Hooker, G.; Koch, P.; Gordo, A.; and Caruana, R. 2023. Considerations when learning additive explanations for black-box models. *Machine Learning*, 112(9): 3333–3359.

Team, L. 2024. Meta Llama Guard 2. https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL_CARD.md.

Templeton, A.; Conerly, T.; Marcus, J.; Lindsey, J.; Bricken, T.; Chen, B.; Pearce, A.; Citro, C.; Ameisen, E.; Jones, A.; Cunningham, H.; Turner, N. L.; McDougall, C.; MacDiarmid, M.; Freeman, C. D.; Sumers, T. R.; Rees, E.; Batson, J.; Jermyn, A.; Carter, S.; Olah, C.; and Henighan, T. 2024. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. *Transformer Circuits Thread*.

Wang, K.; Variengien, A.; Conmy, A.; Shlegeris, B.; and Steinhardt, J. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*.

Wei, A.; Haghtalab, N.; and Steinhardt, J. 2023. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36: 80079–80110.

Wollschläger, T.; Elstner, J.; Geisler, S.; Cohen-Addad, V.; Günnemann, S.; and Gasteiger, J. 2025. The geometry of refusal in large language models: Concept cones and representational independence. *arXiv preprint arXiv:2502.17420*.

Xie, T.; Qi, X.; Zeng, Y.; Huang, Y.; Sehwag, U. M.; Huang, K.; He, L.; Wei, B.; Li, D.; Sheng, Y.; et al. 2024. Sorry-bench: Systematically evaluating large language model safety refusal. *arXiv preprint arXiv:2406.14598*.

Zhao, J.; Huang, J.; Wu, Z.; Bau, D.; and Shi, W. 2025. Llms encode harmfulness and refusal separately. *arXiv preprint arXiv:2507.11878*.

Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.