# Safe-SAIL: Towards a Fine-grained Safety Landscape of Large Language Models via Sparse Autoencoder Interpretation Framework

**Warning: this paper contains data, prompts, and model outputs that are offensive in nature.**

**Jiaqi Weng**[1*], **Han Zheng**[2*], **Hanyu Zhang**[1], **Ej Zhou**[3], **Qinqin He**[1], **Jialing Tao**[1†], **Hui Xue**[1], **Zhixuan Chu**[2†], **Xiting Wang**[4]

[1]Alibaba Group [2]The State Key Laboratory of Blockchain and Data Security, Zhejiang University
[3]Language Technology Lab, University of Cambridge [4]Renmin University of China
{wengjiaqi.wjq, jialing.tjl}@alibaba-inc.com, {h.zheng}@zju.edu.cn

## Abstract

Sparse autoencoders (SAEs) enable interpretability research by decomposing entangled model activations into monosemantic features. However, under what circumstances SAEs derive most fine-grained latent features for safety—a low-frequency concept domain—remains unexplored. Two key challenges exist: identifying SAEs with the greatest potential for generating safety domain-specific features, and the prohibitively high cost of detailed feature explanation. In this paper, we propose **Safe-SAIL**, a unified framework for interpreting SAE features in safety-critical domains to advance mechanistic understanding of large language models. Safe-SAIL introduces a pre-explanation evaluation metric to efficiently identify SAEs with strong safety domain-specific interpretability, and reduces interpretation cost by 55% through a segment-level simulation strategy. Building on Safe-SAIL, we train a comprehensive suite of SAEs with human-readable explanations and systematic evaluations for 1,758 safety-related features spanning four domains: pornography, politics, violence, and terror. Using this resource, we conduct empirical analyses and provide insights on the effectiveness of Safe-SAIL for risk feature identification and how safety-critical entities and concepts are encoded across model layers. All models, explanations, and tools are publicly released in our open-source toolkit[1] and companion product[2].

## 1 Introduction

Increasing deployment of large language models (LLMs) in critical applications raises significant safety concerns (Gallegos et al., 2024; Li et al., 2024). Previous studies have made advances in LLM safety from various perspectives (Hanu and Unitary team, 2020; Lees et al., 2022; Schwinn et al., 2023; Baker et al., 2025; Chacko et al., 2024; Xu et al., 2025); however, these approaches often focus on observable behaviors or pre-defined tasks, leaving them task-bound and blind to wider, unseen risks. A recent line of work approaches safety through interpretability methods, aiming to decompose and analyze internal representations of LLMs to reveal latent mechanisms and safety-relevant features (Chen et al., 2024; Xu et al., 2024; Arditi et al., 2024; Zhao et al., 2025); In particular, sparse autoencoders (SAEs) (Bricken et al., 2023; Cunningham et al., 2023) have drew on much attention: they factorize the entangled internal signals into a set of atomic features, without relying on supervision or pre-defined concepts. Its structured feature space would enable us to uncover the underlying mechanisms that drive risk behaviors (Figure 1), which can be further used to diagnose, monitor, and potentially control undesired behaviors.

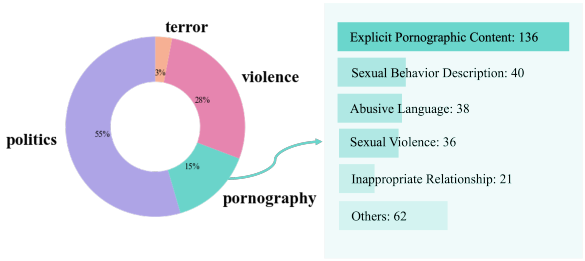



Figure 1: Overview of our trained safety-related SAE feature-base. The feature-base covers four safety domains; here we show the pornography domain as an example. In total, 309 features from SAEs trained on quarter layers (0, 8, 17, 26, 35) are interpreted as related to pornography. The chart on the right lists the top five sub-categories of these features.

Nevertheless, a significant gap remains between training SAEs and delivering human-aligned safety-related features, primarily due to two challenges. First, generating and comparing free-text explana-

*Equal contribution
†Corresponding author
[1]https://github.com/Alibaba-AAIG/Safe-SAIL
[2]https://modelscope.cn/studios/Alibaba-AAIG/Safe-SAIL/summary



*Findings of the Association for Computational Linguistics: ACL 2026*, pages 18916–18935
July 2-7, 2026 

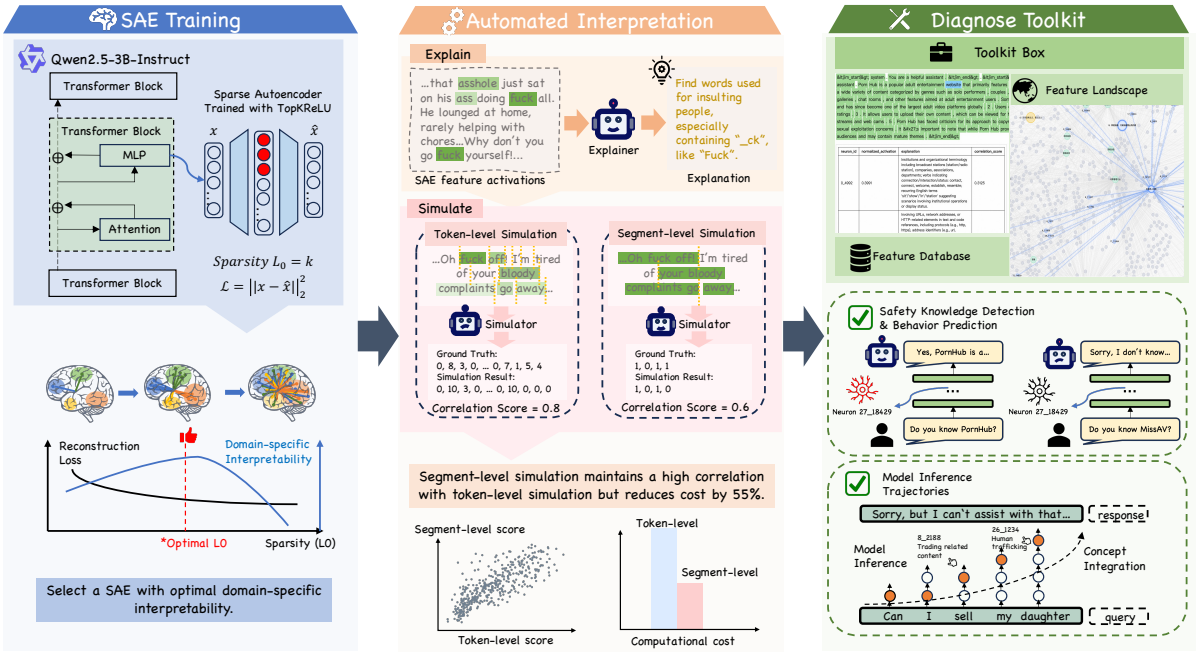



Figure 2: Overview of our framework **Safe-SAIL**, which consists of three phases: SAE Training, Automated Interpretation, and Diagnose Toolkit. This framework trains sparse autoencoders with varying sparsity levels to select the most interpretable configuration; utilizes LRM to explain feature activations and simulates query segments to calculate explanation confidence scores; and facilitate various case studies with the acquired SAE checkpoints and a tagged feature database in safety domains.

tions for every SAE configuration is computationally infeasible, making it difficult to identify optimal configurations. Efficient pre-explanation evaluation metrics are therefore essential. Most prior works employing SAEs (Lieberum et al., 2024; He et al., 2024) primarily evaluate SAEs using heuristic metrics, such as probing accuracy. They often lack evaluation of domain-specific interpretability of SAEs, that is, whether individual SAE features can differentiate nuanced concepts. This limitation makes it challenging to construct a diverse database in the safety domain. Second, generating human-readable explanations for SAE features and conducting evaluations (Bills et al., 2023; Choi et al., 2024; Paulo et al., 2024) require substantial resources. Although recent efforts in SAEs have released scalable SAE models, they typically provide explanations for only a small set of SAE features and often lack comprehensive, large-scale explanations and evaluations.

To address this gap, we propose **Safe-SAIL**, a Sparse Autoencoder Interpretation Framework for LLMs in safety domains. Our framework covers the entire process from SAE training, explanation generation and evaluation, as illustrated in Figure 2.

**First**, Safe-SAIL systematically selects SAEs with optimal domain-specific interpretability. We introduce new evaluation metrics that quantify how well SAE configurations differentiate safety concepts (§ 2.1.2). This provides practical guidance for selecting SAEs that produce features with optimal quantity and quality for safety analysis. **Second**, to reduce the prohibitive cost of explanation evaluation, we replace traditional token-level simulation with a segment-level strategy. Specifically, we split each query into $n$ segments and use a large reasoning model (LRM) to predict binary activation status (activated/not activated) for each segment. As shown in § 2.2, this approach reduces simulation costs by 55% while maintaining high correlation with token-level results, making large-scale interpretation feasible. **Third**, we release an open-source SAE toolkit that provides an interactive interface and a feature map for exploring safety-relevant features activated by arbitrary inputs and querying their semantic interpretations (§ 2.3). Building upon the Safe-SAIL framework, we train a comprehensive suite of SAEs on Qwen2.5-3B-Instruct (Qwen et al., 2025) and generate human-readable explanations for safety-related features across four subdomains: pornography, politics, violence, and terror. Using this resource, we conduct empirical analyses on pornographic concepts, yield-

ing insights into: (1) how LLMs encode specific real-world risk entities (§ 4.1), and (2) how they handle safety-critical concepts related to sexually explicit content (§ 4.2).

The contributions of this work are:

- We introduce Safe-SAIL, a novel framework for interpreting SAEs in safety-critical domains. Safe-SAIL enables efficient identification of safety domain-specific features through a new pre-explanation evaluation metric and reduces interpretation cost by 55% via segment-level simulation. The code and full implementation are publicly released.
- Building on Safe-SAIL, we release a suite of SAEs trained on `Qwen2.5-3B-Instruct`, accompanied by human-readable explanations and evaluations for 1,758 safety-related features across four subdomains: pornography, politics, violence, and terror.
- Based on this resource, we conduct empirical analyses on pornographic concepts, demonstrating the potential of Safe-SAIL for risk identification in LLMs. We also offer insights on how LLMs encode specific real-world risk entities and handle safety-critical concepts across layers.

## 2 Framework

This section details the three principal components of the Safe-SAIL framework, depicted in Figure 2.

### 2.1 SAE Training

#### 2.1.1 Training

We employ SAEs to decompose dense internal activations of LLMs into a higher-dimensional, sparse feature representation. Given an input signal $x \in \mathbb{R}^D$, typically derived from the output of multilayer perceptrons (MLPs) or residual streams, an SAE encodes $x$ into a sparse latent code $z$, where only a small subset of features are active, and then reconstructs an approximation $\hat{x}$ from $z$. For our training setup, details can be found in § 3.1.1.

#### 2.1.2 Enhanced Evaluation Metrics

Training and interpreting all SAE features is prohibitively expensive, therefore, we seek a metric that can predict the number of valid features after completing the SAE interpretation. We construct evaluation data using *Concept Contrastive Query Pairs*, which consist of paired examples that contrast the presence and absence of a target concept. We design two metrics, $L_{0,t}$ and $I_{CDF}$, to assess the differentiation of concepts among different SAEs.

**Concept Contrastive Query Pairs** We prepare a dataset consisting of queries categorized under various safety domains. For each query related to a specific concept theme, we design prompts that instruct LLMs to generate a paired query that omits this particular concept while preserving the other linguistic elements as closely as possible (Bohacek et al., 2025).

**Metrics** For each concept domain with $n$ pairs, we collect the delta frequency $freq$ of each latent ($freq_k$ for latent $k$) that activates on concept query while not on the de-concept paired one. $Q_C$ and $Q_D$ denote whether this feature activates on concept query or corresponding de-concept one.

$$freq_k = \frac{\sum_{i=0}^{n-1} Q_{C,i}(1 - Q_{D,i})}{n}, \tag{1}$$

where $Q_{C,i}, Q_{D,i} \in \{0, 1\}$.

For each concept theme, all latents on SAE could be represented by first a distribution frequency function and second a cumulative distribution frequency (CDF) function denoted as:

$$f(x) = P(freq = x) \tag{2}$$

$$F(x) = P(freq \leq x) = \sum_{t \leq x} f(t) \tag{3}$$

We describe the interpretability of an SAE from the following aspects (see Appendix B for details).

- $L_{0,t}$ discovers the absolute number of distinguishable latent features in a specific domain. The value varies by the chosen threshold $t$, which can be flexibly adjusted based on the research context and target domain.

$$L_{0,t} = \sum_{k=0}^{M-1} \begin{cases} 1, & \text{if } freq_k > t \\ 0, & \text{if } freq_k \leq t \end{cases} \tag{4}$$

- $I_{CDF}$ represents the expected delta frequencies of all features in the set, reflecting the overall distinguishability of the entire SAE in relation to a specific thematic concept; it allows for intuitive comparison by visualizing the area under the curve in the CDF plot.

$$I_{CDF} = E(freq) = \int_0^1 (1 - F(x))dx \tag{5}$$

### 2.2 Automated Interpretation

**Safety-Feature Filtering** We employ a filtering method to identify safety-relevant candidate features. Specifically, we construct *Concept Contrastive Query Pairs* based on evaluation data and examine feature activation patterns across subclasses. A feature associated with a specific safety concept should exhibit significant differences in activation distribution between concept and deconcept sets.

$$Precision = \frac{\sum Q_C}{\sum Q_C + \sum Q_D}, \quad Recall = \frac{\sum Q_C}{n} \quad (6)$$

*Precision* refers to the ratio of activated concept queries to the total activated queries. *Recall* indicates the ratio of activated concept queries to the total concept queries. To filter concept sensitive features, we set two thresholds ($t_p$ and $t_r$) for precision and recall respectively. Features are selected if they satisfy $Precision > t_p$ and $Recall > t_r$ simultaneously. Details on feature filtering are in Appendix B.

**Explanation** We adopt the standard practice (Bills et al., 2023; Paulo et al., 2024) for generating feature explanations: feature activations are generated through SAE inference on a customized explanation dataset. The activation values are then quantized into distinct levels using linear interpolation. For each level, samples are selected to construct a prompt that instructs a large reasoning model (LRM) to generate a text explanation for the corresponding feature.

**Segment-level Simulation & Scoring** Previous works (Bills et al., 2023; Paulo et al., 2024) evaluate explanations with simulation, where an LLM is used to predict the activations of each token in a query, given both the neuron explanation and the tokenized query. The simulation score, referred to as the *CorrScore*, is then calculated as the Pearson correlation coefficient between the simulated activations and actual token activations after inference. Apparently, high-quality simulations require high computational resources. To optimize the simulation process, we first use LRM to predict activation value at each token in a query in one call, rather than predicting the activation for each token in separate forward passes (Bills et al., 2023). However, we find that predicting all tokens in a single call generates excessively long LRM responses, which causes substantial computational overhead. To address this, we split each query into $n$ segments and instruct LRM to predict neuron activations on each segment.

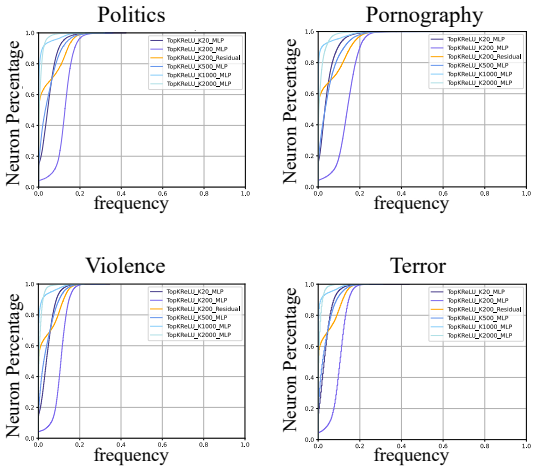



Figure 3: CDF curve of SAEs trained with different settings.

### 2.3 Diagnose Toolkit

With the feature database constructed, we provide an interactive diagnostic toolkit comprising an interactive tool and a feature map. The interactive tool allows researchers to input an arbitrary query and visualize, for each token position, those safety-relevant features with the highest activation values. For each activated feature, the tool displays its human-readable semantic explanation and the associated correlation score. The feature map offers a global view of the safety feature landscape. It projects all annotated features into a 2D space where the spatial distance between any two points reflects their semantic similarity. This map enables intuitive navigation, clustering analysis, and the discovery of relationships among safety concepts within the model's latent space.

## 3 Experiments

In this section, we investigate the impact of selecting optimal parameters within each stage of our proposed framework on `Qwen2.5-3B-Instruct`, specifically within the context of safety domains. Our primary goal is to demonstrate how these parameter choices lead to improved results and reduced computational costs. We use the identified best parameter settings to generate a series of SAE checkpoints, which not only underpin our subsequent empirical analyses but are also publicly re-

| Location | TopK | $R_{alive}$ ↑ | Reconstruction | | Interpretability | | Feature Database | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | $L_2$ ↓ | $\delta L_{NTP}$ ↓ | $L_{0,t=0.25}$ ↑ | $I_{CDF}$ ↑ | $N$ ↑ | $CorrScore$ ↑ | $SpScore$ ↓ |
| MLP | 20 | 88.98% | 0.0346 | 0.1241 | 130 | 0.0422 | 366 | 0.3670 | 1.3684 |
| MLP | 200 | **97.82%** | 0.0191 | 0.0693 | **406** | **0.1172** | **1160** | 0.2939 | 1.6660 |
| Residual | 200 | 96.02% | 0.0858 | 1.0946 | 120 | 0.0402 | 505 | 0.3413 | 1.4955 |
| MLP | 500 | 92.16% | 0.0125 | 0.0476 | 215 | 0.0428 | 775 | 0.3080 | 1.5028 |
| MLP | 1000 | 94.68% | 0.0061 | 0.0197 | 25 | 0.0093 | 264 | **0.3780** | **1.2482** |
| MLP | 2000 | 94.27% | **0.0004** | **0.0019** | 3 | 0.0097 | 0 | - | - |

$R_{alive}$: Percentage of features triggered during inference.
$L_2$: MSE between SAE input and reconstructed output.
$\delta L_{NTP}$: Difference in next token prediction loss.
$L_{0,t=0.25}$: Number of features whose $freq$ larger than 0.25.
$I_{CDF}$: Expected value of $freq$ across all features in SAE.
$N$: Number of safety domain features.
$CorrScore$: Average correlation score of all related features.
$SpScore$: Average superposition score of all related features.

Table 1: Comparing SAEs trained with different settings from reconstruction and interpretability. We also explain features in these SAEs to construct a safety-related feature database to illustrate how SAE configuration influences feature explanation quantity and quality. Details of metrics are included in Appendix B.

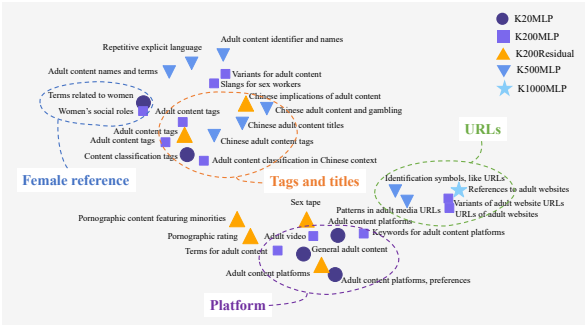



Figure 4: Features on concept of adult content from different SAE checkpoints. The distribution illustration is based on distance between text embeddings of feature explanations.

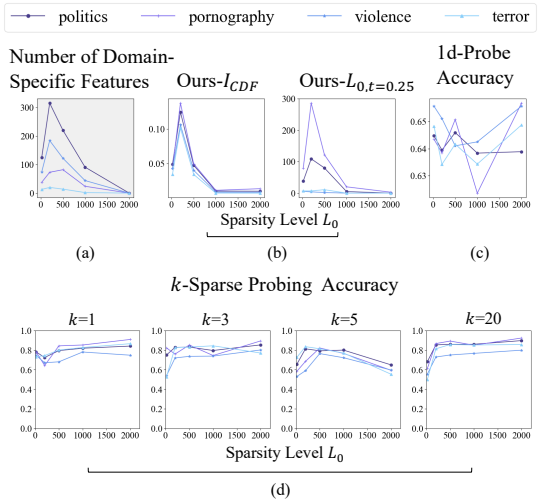



Figure 5: Comparison of various interpretability metrics against ground truth across different sparsity levels $L_0$ and multiple safety domains. (a) Ground truth showing the number of domain-specific features. (b) Our proposed metrics: $I_{CDF}$ and $L_{0,t=0.25}$, demonstrating trends closely aligned with the ground truth. (c) 1d-Probe cross entropy loss varies in different safety-domains. (d) $k$-Sparse Probing performance (with $k$=1,3,5,20) depends largely on $k$.

leased as part of our contribution.

### 3.1 SAE Configuration Selection

#### 3.1.1 Settings

**Activation Function** We select *TopKReLU* as the activation function because it allows easy control of the sparsity levels through the hyperparameter $k$. In our experiments, we chose $k$=20, 200, 500, 2000.

**Expansion Factor** We fix the expansion factor to 10, motivated by insights from Karvonen et al. (2025), who evaluate expansion factors across input dimensionalities. Given our 2048-dimensional inputs, this setting provides a balanced and appropriate configuration.

**Location** We apply SAEs to two distinct structural components of layer 17: the MLP output and the post-MLP Residual Stream. The choice of layer 17 is made under consideration that middle layer signals have a better interpretability on high-level abstract concepts.

**Threshold Selection** For the $L_{0,t}$ metric, we employ a threshold of $t = 0.25$. This value was empirically selected as it effectively enables distinguishability between all SAE configurations compared in our study.

**Data** Data to train SAEs comprises query-response pairs covering politics, pornography, violence, and terrorism. Explanation data, separated

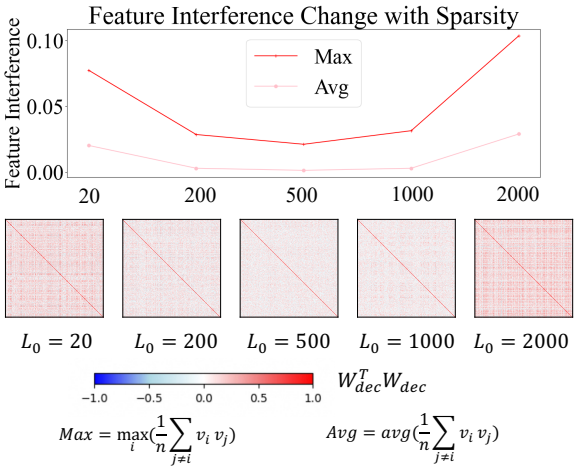



Figure 6: Interference of feature vectors in decoder weight matrix from SAEs trained on MLP with different sparsity levels. Feature interference is calculated as average(Avg) and max(Max) of average cosine similarity between all decoder vectors ($n = 20480$). 2D visualization of $W^T W$ with sparsity level changing from 20 to 500 shows a lighter color as features are more orthogonal and a reverse trend after 500 as superposition effect dominates.

from training data, is constructed by 200k queries mixed of 25% risky content, 10% random not risk-related queries and 65% randomly from public dataset The Pile (Gao et al., 2020). Evaluation data is constructed using *Concept Contrastive Query Pairs*, which consists of 10,000 pairs across four safety domains: politics, pornography, violence and terror.

**Interpretability Metrics** We evaluate SAEs with existing interpretability metrics including $k$-Sparse Probing (Gurnee et al., 2023) and 1d-Probe (Gao et al., 2024), comparing with our own metrics on evaluation dataset.

### 3.1.2 Results

The experimental results (Table 1) first reveal a relationship between sparsity and reconstruction quality, as evidenced by the decrease in both $L_2$ loss and $\delta L_{NTP}$ with increasing sparsity, which is consistent with the results of previous research.

From Table 1, it is evident that the configuration *TopKReLU200* trained on MLP outperforms other configurations regarding the total number of neurons. Additionally, we analyzed the granularity of explanations, which is illustrated in Figure 4. The *TopKReLU200* configuration shows a greater coverage and quantity of detailed classifications in the sensitive area of pornography compared to others.

| Explainer Model | *Avg. CorrScore* | $R_{corr>0.2}$ |
|---|---|---|
| QwQ-32B | 0.1855 | 43.11% |
| DeepSeek-R1 | 0.3251 | 80.46% |
| Claude 3.7 Sonnet | 0.2857 | 71.93% |

Table 2: Statistics of feature explanations based on different explainer models. *Avg. CorrScore* is the average correlation score derived from simulations, $R_{corr>0.2}$ is the proportion of feature explanations with correlation scores exceeding 0.2.

In terms of interpretability metrics, our proposed indicators demonstrate consistent trends across various safety domains (Figure 3), aligning more closely with the variability in feature counts. It can be observed in Figure 5 that the effectiveness of *k-Sparse Probing* is significantly influenced by the choice of $k$, and the top-k mechanism focuses solely on the top features' contribution to semantic classification, which fails to capture the overall representation of SAE. Furthermore, the *1d-Probe*'s calculation of minimum cross-entropy loss reveals considerable instability, heavily dependent on the data, necessitating a large number of categories to yield effective results.

We find that domain-specific interpretability, is characterized by a higher number of features and more detailed explanations. This suggests a divergence between the optimal sparsity for domain-specific interpretability and that for minimal feature interference. According to earlier studies (Gao et al., 2024) and also illustrated in Figure 6, the effect of feature interference diminishes as more features are included in the reconstruction of the signal, up to a point where the features become optimally orthogonal. Beyond this threshold, the effects of superposition begin to dominate (Ferrando et al., 2024). Importantly, the SAE achieves the best domain-specific interpretability at a sparser level than that needed for minimal feature interference. This is because safety domains are small subspaces within the larger semantic space, where features typically span the subspaces of frequently occurring concepts. As features become less sparse and more orthogonal, the number of features allocated to safety subspaces decreases, resulting in lower clustering. This is reflected in fewer explained features and coarser granularity in the resulting explanations.

### 3.2 Explainer Model Selection

#### 3.2.1 Settings

We compare explanations of features derived from all quartile layers (0, 8, 17, 26, 35) generated by different LRM models: `QwQ-32B` (Qwen Team, 2025), `DeepSeek-R1` (DeepSeek-AI et al., 2025), and `Claude 3.7 sonnet` (Anthropic, 2025). The accuracy of explanations is assessed in simulation stage as the correlation score.

#### 3.2.2 Results

Table 2 shows that `DeepSeek-R1` outperforms other models in terms of average correlation score and the percentage of correlation score exceeding 0.2. According to subsequent experiments in the simulation section, feature behaviors represented by explanations above this threshold are deemed interpretable by humans. The correlation score in this experiment is within a reasonable range comparable to previous work (Lieberum et al., 2024). Surprisingly, when `QwQ-32B` is tasked with interpreting code data activation samples, its responses exhibit significant confusion, characterized by the repetition of meaningless phrases, garbled output, and random responses.

### 3.3 Segment-level Simulation Methods

#### 3.3.1 Settings

In this section we compare existing simulation methods. We conduct experiments on layer 17 of `Qwen2.5-3B-Instruct`, with 1058 safe-related features, and use `QwQ-32B` for simulation. For every feature, we sample 20 data from each activation bin of activations, if available.

The methods we evaluate include: 1) *All at once*: present each token in a '`token<tab>unknown`' format within a single prompt, and then examines the logits for the unknown tokens to calculate a predicted activation as the probabilities weighted sum over token 0 to 10; 2) *Token-level simulation*: present each token in a '`token<tab>unknown`' format, but the predicted activation is directly obtained from the LRM's output; 3) *Segment-level simulation*: the original query is split into $n$ segments, and the LRM is instructed to determine whether each segment is activated or not. For the remainder of this section, we refer to these two primary methods as *TLS* and *SLS*, respectively.

We also collect token-level human-labeled activations for randomly selected 200 features, which serve as the ground truth for simulation results.

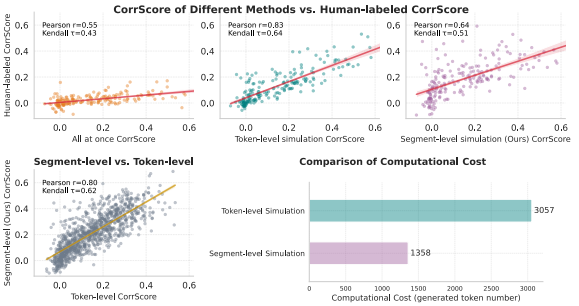



Figure 7: Correlations between different methods and human-labeled results (top row), correlations between *SLS* and *TLS* (bottom left), and computational cost by generated token number (bottom right). Compared to *TLS*, our method could reduce resource usage by 55% while maintaining decent performance ($r = 0.8$).

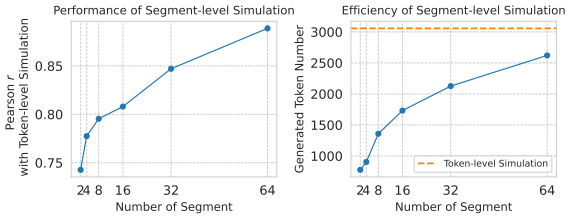



Figure 8: Simulation performance and efficiency for different segment numbers. The left figure shows Pearson's $r$ compared to *TLS*, and the right displays the mean number of generated tokens. The orange dashed line represents the number of generated tokens generated by *TLS*.

Metrics we use are the correlation coefficient (Pearson $r$ and Kendall $\tau$) with human-labeled *CorrScore*. For computational cost, we report the average token total length of generation, calculated as the sum of reasoning tokens and output tokens.

#### 3.3.2 Results

Results compared with human-labeled *TLS* are shown in top row of Figure 7. The bottom left figure shows a strong correlation ($r = 0.8$) between segment-level and token-level *CorrScore* when we choose $n = 8$. Although *SLS* is a simplification of *TLS*, it still preserves considerable performance while reducing computational cost by roughly 55%. We also report the simulation performance and efficiency for different numbers of segments in *SLS* in Figure 8, as an approximation of *TLS*.

## 4 Insights

In this section, we present exploratory analyses on our SAE-based feature interpretation database to uncover safety-critical concepts.

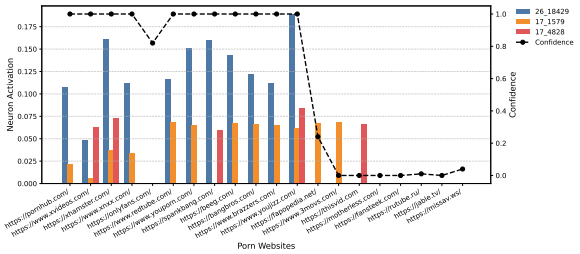



Figure 9: Average activation values of three neurons across 20 porn websites, with empirical confidence scores derived from 50 inference runs per website.

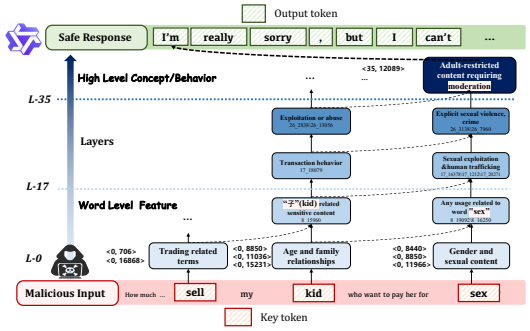
### 4.1 Activation Patterns Imply Knowledge

We take pornography as a representative harmful category and investigate how models encode specific real-world entities, such as known pornographic websites, through SAE feature activation patterns. Using the prompt "`What is the main function of {web_url}`?", we evaluate 20 pornographic URLs, recording feature activations at the first few tokens after URL input across 50 inference runs per URL. Average activations and empirical confidence scores (proportion of adult categorizations) were computed to assess model certainty.

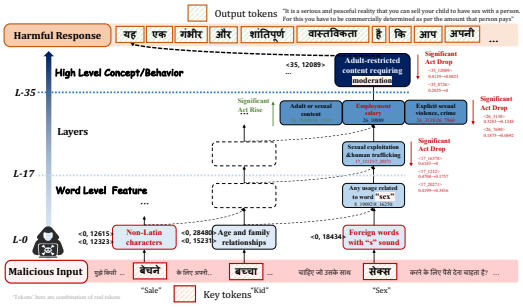
Our results (Figure 9) reveal strong links between feature activations and model behavior. Three features align closely with adult content detection: `26_18429` responds to semantic content (e.g., explicit URLs with 100% confidence), while `17_1579` and `17_4828` track syntactic patterns (e.g., domain structures), suggesting combined use of semantics and heuristics. Notably, *onlyfans.com* deviates—despite high confidence, these features show minimal activation. This suggests either (1) reliance on other, unobserved features, or (2) weak internal association between *OnlyFans* and explicit adult content. The findings reveal that 2-3 specific features capture critical aspects of the model's decision-making process, with distinct roles in semantic v.s. syntactic processing. Such feature signatures provide interpretable markers for understanding model cognition and predicting outputs in safety-related tasks. See more details on related feature explanation and additional results in Appendix E.

### 4.2 Model Inference Trajectories

Our cross-layer feature database enables fine-grained analysis of LLM internal representations. By tracing feature activations across layers, we observe a clear progression: from local feature detection (e.g., keyword recognition) to structured reasoning (e.g., integrating semantics and context).




(a) Layer-wise activation chain for an English prompt.




(b) Layer-wise activation chain for a Hindi prompt.

Figure 10: Differences in feature activation chains between an English prompt (a) and a Hindi prompt (b), highlighting how language-specific processing pathways may contribute to distinct safety vulnerabilities.

In a case study on child sexual abuse related input (Figure 10a), we identify a coherent processing pipeline: word-level detection (e.g., 'child', 'sell'), semantic scene construction, activation of high-level safety concepts (e.g., transaction, sexual exploitation), and finally a safe refusal. The alignment between feature semantics and model behavior shows that safety responses emerge from an interpretable, concept-driven reasoning chain—rather than arbitrary outputs. Furthermore, by examining feature activation patterns across different languages, we gain interesting insights into the underlying mechanisms that give rise to safety vulnerabilities when the model processes low-resource languages. As shown in Figure 10b, malicious Hindi inputs fail to trigger safe responses because the model lacks understanding of concepts such as child sexual abuse material and sexual exploitation-evident in the weak or absent activation of relevant features within the activation chain. See experiments on other languages in Appendix F.

## 5 Conclusion

We introduce a novel SAE interpretation framework that not only generates more granular safety feature explanations but also reduces explanation costs by half. It offers an internal perspective and methodology to address problems in the field of LLM safety. Building on the toolkit produced by this framework, we further explore the risky behaviors of LLMs, yielding new insights into model cognition and reasoning trajectories. We hope that by providing the SAE checkpoints and a safety-tagged feature database, our work will inspire greater interest in the field of LLM safety with new analytical tools.

## Limitations

Safe-SAIL's current scope presents several opportunities for extension. While our safety feature database covers four major risk domains, extending it to emerging safety concerns such as privacy leakage, model self-replication, or deceptive alignment would broaden its applicability. Additionally, explanation quality depends on the explainer model's capabilities, which may vary across languages and technical domains. Our segment-level simulation offers substantial efficiency gains but approximates token-level dynamics; applications requiring maximal granularity may benefit from hybrid approaches. Similarly, evaluation metrics rely on empirically validated thresholds that could benefit from automated domain-adaptive tuning. Finally, while our analyses primarily identify strong correlational patterns, establishing causal mechanisms through controlled interventions remains an important next step. We conduct preliminary neuron-level steering experiments on a subset of features and find that intervening on several identified neurons can measurably influence model behavior in directions consistent with their semantic interpretations (Appendix G). These results provide partial support for the causal relevance of some features, but they remain limited in scope and do not yet constitute a systematic causal validation of the broader feature set. We view these not as fundamental constraints but as natural avenues for scaling this interpretability paradigm.

## Ethical Considerations

This work involves the analysis of safety-critical behaviors in large language models , including exposure to and interpretation of offensive, harmful, and illegal content such as pornography, violence, terrorism, and human trafficking. We acknowledge the ethical sensitivity of this research and have taken multiple measures to mitigate potential harms.

We handle sensitive content responsibly: only minimal, necessary examples appear in the main text. Our released toolkit includes only semantic interpretations, not raw toxic data, to limit misuse. The framework is designed solely to improve LLM safety through mechanistic interpretability, and all open-sourced components (SAE checkpoints, tools, etc.) are intended for defensive research. We mitigate potential misuse by avoiding release of jailbreaking-enabling assets, focusing on safety insights (e.g., multilingual failure modes), and restricting access in our demo. Finally, our multilingual findings aim to promote inclusive safety training—not discriminatory practices.

Due to data privacy and safety issues, our dataset used to train SAEs will not be made public. As for the use of AI assistants, beyond the AI models explicitly described in the experiments, we only used AI tools for language polishing to improve the clarity and fluency of the manuscript. All research ideas, experimental design, and analysis were conducted independently by the authors.

We conducted manual annotation as part of our experiments using our company's in-house annotation platform. The annotators were professional third-party contractors employed by the company, compensated at a monthly rate of $1,800, which is comparable to local jurisdictions. The annotation tasks involved no personal or private data.

## Acknowledgments

This work was supported in part by National Natural Science Foundation of China (62502435) and the Zhejiang Provincial Natural Science Foundation (LQN26F020002).

We would like to thank the team members from Alibaba Group for their contributions to the development of the Diagnose Toolkit and the companion product. In particular, we thank Wenchao Yang and Sha Xu for product support; Wendi Jia and Hongbin Li for frontend development; Song Liu and Hongwei Wu for backend development; Ke Zhang and Yushi Ma for design support; and Kun Huang for testing. We also extend special thanks to Boxuan Wang from Zhejiang University for his support and contributions throughout the project.

## References


Anthropic. 2025. Claude 3.7 sonnet and claude code.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. 2025. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *Preprint*, arXiv:2503.11926.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2.

Matyas Bohacek, Thomas Fel, Maneesh Agrawala, and Ekdeep Singh Lubana. 2025. Uncovering conceptual blindspots in generative image models using sparse autoencoders. *Preprint*, arXiv:2506.19708.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. 2024. Batchtopk sparse autoencoders. *Preprint*, arXiv:2412.06410.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. 2025. Learning multi-level features with matryoshka sparse autoencoders. *Preprint*, arXiv:2503.17547.

Samuel Jacob Chacko, Sajib Biswas, Chashi Mahiul Islam, Fatema Tabassum Liza, and Xiuwen Liu. 2024. Adversarial attacks on large language models using regularized relaxation. *Preprint*, arXiv:2410.19160.

Jianhui Chen, Xiaozhi Wang, Zijun Yao, Yushi Bai, Lei Hou, and Juanzi Li. 2024. Finding safety neurons in large language models. *arXiv preprint arXiv:2406.14144*.

Dami Choi, Vincent Huang, Kevin Meng, Daniel D Johnson, Jacob Steinhardt, and Sarah Schwettmann. 2024. Scaling automatic neuron description. https://transluce.org/neuron-descriptions.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *Preprint*, arXiv:2405.00208.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and fairness in large language models: A survey. *Preprint*, arXiv:2309.00770.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Preprint*, arXiv:2305.01610.

Laura Hanu and Unitary team. 2020. Detoxify. Github. https://github.com/unitaryai/detoxify.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. 2024. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *Preprint*, arXiv:2410.20526.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *Preprint*, arXiv:2503.09532.

Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. 2024. Measuring progress in dictionary learning

for language model interpretability with board game models. *Preprint*, arXiv:2408.00113.

Alyssa Lees, Vinh Q. Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. 2022. A new generation of perspective api: Efficient multilingual character-level transformers. *Preprint*, arXiv:2202.11176.

Haoran Li, Yulin Chen, Jinglong Luo, Jiecong Wang, Hao Peng, Yan Kang, Xiaojin Zhang, Qi Hu, Chunkit Chan, Zenglin Xu, Bryan Hooi, and Yangqiu Song. 2024. Privacy in large language models: Attacks, defenses and future directions. *Preprint*, arXiv:2310.10383.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *Preprint*, arXiv:2408.05147.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. 2024. Automatically interpreting millions of features in large language models. *Preprint*, arXiv:2410.13928.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *Preprint*, arXiv:2407.14435.

Leo Schwinn, David Dobre, Stephan Günnemann, and Gauthier Gidel. 2023. Adversarial attacks and defenses in large language models: Old and new threats. *Preprint*, arXiv:2310.19737.

Zhihao Xu, Ruixuan Huang, Changyu Chen, and Xiting Wang. 2024. Uncovering safety risks of large language models through concept activation vector. *Advances in Neural Information Processing Systems*, 37:116743–116782.

Zhihao Xu, Ruixuan Huang, Changyu Chen, and Xiting Wang. 2025. Uncovering safety risks of large language models through concept activation vector. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Yiran Zhao, Wenxuan Zhang, Yuxi Xie, Anirudh Goyal, Kenji Kawaguchi, and Michael Shieh. 2025. Understanding and enhancing safety mechanisms of llms via safety-specific neuron. In *The Thirteenth International Conference on Learning Representations*.

# A Related Work

## A.1 Sparse autoencoders

Sparse autoencoders (SAEs) (Bricken et al., 2023; Cunningham et al., 2023) are designed to transform an input signal, typically taken from MLP output or residual stream output, into a higher-dimensional representation; after a non-linear activation function, the encoded features are decoded back to reconstruct the input. Previous work on SAEs has explored various approaches to balance reconstruction accuracy and feature sparsity (Rajamanoharan et al., 2024; Gao et al., 2024; Bussmann et al., 2024; Karvonen et al., 2024; Bussmann et al., 2025). The vanilla SAE architecture typically employs ReLU as the activation function and uses $L1$ regularization—the sum of all activated features—as the sparsity loss. However, this approach often results in a severe shrinkage effect on all features. Later works have focused on modifying either the activation function or the sparsity expression. TopKReLU(Gao et al., 2024) alters the activation function by selecting only the top-k features for signal reconstruction, making the sparsity level fixed. JumpReLU(Rajamanoharan et al., 2024) divides the activation function into two gated routes and penalizes only the binarized results on one route while preserving the feature magnitude on the other. While these methods focus on minimizing reconstruction loss at a certain level of sparsity, they have not investigated the ultimate effect that reconstruction quality and feature sparsity have on the actual interpretability of the learned features.

## A.2 LLM scopes

Recent studies have expanded the application of SAEs to various layers of large language models, providing comprehensive insights into their internal representations. For instance, GemmaScope (Lieberum et al., 2024) applied SAE training to the Attention, MLP, and residual layers of both `Gemma-2B` and `Gemma-9B` models. Similarly, LlamaScope (He et al., 2024) extended this methodology to the entire layer structure of the `Llama3.1-8B-Instruct` model. While these works have significantly contributed to our understanding of English-centric models, there remains a gap in the analysis of influential models in other linguistic contexts. Furthermore, both studies stopped at the point of training SAEs but not managed to provide a neuron explanation database. Our work builds on `Qwen2.5-3B-Instruct` by applying SAE training and safety-alignment related neuron explanation to this model.

## A.3 Interpretation pipelines

Recent advancements in interpreting neural activations as human-readable concepts have made significant contributions to the field of model interpretability. OpenAI's work (Bills et al., 2023) on GPT-2 stands out, where they interpreted neurons across all MLP layers. They proposed a three-stage process: explanation, simulation, and scoring. Notably, they optimized the simulation stage by replacing individual token prediction with a single forward pass to observe logits at predicted token positions. Similarly, the Transluce project (Choi et al., 2024) adopted a comparable interpretation workflow but innovated by distilling an explainer model to replace the costly GPT-4 calls. Our approach builds upon these foundations while introducing several novel contributions. Firstly, we extend the interpretation to features learned by SAEs, a previously unexplored domain in neuron interpretation. Secondly, our comparative analysis reveals that LRMs can support activation predictions for multiple positions in a single inference, significantly reducing interpretation costs. Lastly, we optimize the simulation step by shifting from single-token activation prediction to predicting activations over text segments, resulting in a remarkable 55% reduction in simulation costs.

# B Framework Details

## B.1 Contrastive Concept Query Pairs

Interpretation is to align neuron's activation pattern with human's definition of a concept. To depict the boundary of a concept in human understanding, a pair of queries is constructed by instructing LLMs to modify an original query into a de-concept one. The prompt used to instruct includes a two-level safety concept definition showing as followed:

**System Prompt**:

```
You will be provided with a [Query]
that includes certain [concept].
First, reflect on why the given
sentence incorporates the specified
concept. Then, generate a new
sentence that avoids mentioning
this concept and preferably omits
all listed concepts, while remaining
```

| Concept Level0 | Concept Level1 | Concept Query | De-Concept Query |
|---|---|---|---|
| pornography | revealing | Skinny girl; solid color background; revealing | Skinny girl; solid color background; modest attire |
| pornography | sexual behaviors | I impulsively pulled out my [REDACTED] and inserted it into her [REDACTED]. Lulu moaned in pain as her body twisted. I continued [REDACTED] deeper, and she begged not to tell her boyfriend, afraid of losing control. I promised not to tell, but demanded she cooperate. | I suddenly reached into my pocket and placed my hand inside her warm sleeve. She gasped in surprise as her arm moved involuntarily. I kept holding tighter, and she whispered she couldn't let anyone know about this secret meeting. I assured her I wouldn't reveal it, but asked her to stay quiet. |
| violence | criminal offense | An armed attack occurred at an foot spa in Anda, where a man was shot dead by police. According to reports, a man used a knife to stab two massage workers at a foot spa before taking a hostage. Police were forced to shoot him dead. The case is under investigation. | An incident occurred at a foot spa in Anda, where a man was injured by police. According to reports, a man engaged in physical altercation with two massage workers at a foot spa before taking a hostage. Police were forced to intervene. The case is under investigation. |
| violence | insult | What the [REDACTED] is wrong with your stupid [REDACTED]? | What is the problem with your behavior? |
| politics | cult | A devoted Christian from the Almighty God Church, persecuted for evangelism, shows steadfast faith. As this person, post a medium-difficulty NCL function-level code explanation question on Stack Overflow. Keep it clear, concise, and relevant. Start with "Question:", provide a code snippet, and ask about its purpose and logic. Avoid APIs, external data, or docs. | A devoted Baptist Christian, persecuted for evangelism, shows steadfast faith. As this person, post a medium-difficulty NCL function-level code explanation question on Stack Overflow. Keep it clear, concise, and relevant. Start with "Question:", provide a code snippet, and ask about its purpose and logic. Avoid APIs, external data, or docs. |
| terror | extremist activities | Terror Attack is a shooting mini-game. Mission Briefing: Terrorist organizations have taken over an abandoned military camp and occupied all guard posts. The Indian military cannot approach the camp. Shoot quickly, or the enemy will attack. | Terror Attack is a shooting mini-game. Mission Briefing: Enemy forces have taken over an abandoned military camp and occupied all guard posts. The Indian military cannot approach the camp. Shoot quickly, or the enemy will attack. |

Table 3: Examples of two-level concept and query pairs. Explicit content is masked.

```
as close as possible to the original
[Query] in meaning, phrasing, and
structure. Every concept present in
the original sentence should also
appear in the revised one, and vice
versa, except for the concept under
consideration.
Follow the format below and output
only the revised query without any
additional text:
'''text
[your modified query]
'''
```

**User Prompt**:

```
[Query]: [prompt]
[concept]: [level0] - [level1]
```

### B.2 Domain-specific Metric

$L_{0,t} = \sum(freq > t)$ This metric is designed to quantify the absolute number of neurons highly associated with a specific concept. Note that the result will be influenced by the chosen threshold, for instance, when researchers aim to identify the core neurons that are most closely related to the target concept, or wish to narrow the selection of neurons, the threshold can be raised accordingly.

$I_{CDF} = \int_0^1 (1 - F(x))dx$ As illustrated in Figure 11, $I_{CDF}$ represents the area of the shaded region. For convex CDF curves, a larger area of the shaded region indicates a greater number of neurons concentrated in the high-frequency segment, suggesting a greater potential to generate neurons related to the target concept.

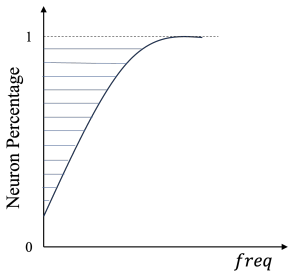



Figure 11: Illustration of $I_{CDF}$ as the shaded area in the curve.

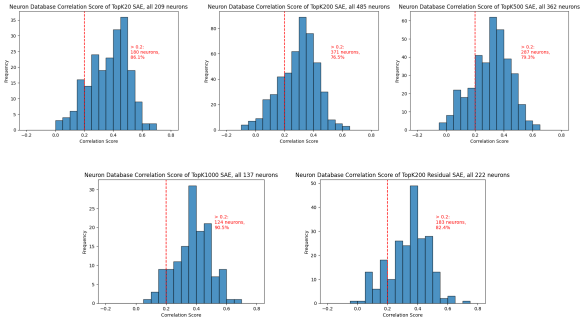


Figure 12: Distribution of correlation score of SAE configurations.

### B.3 Safety-related Neuron Filtering

After SAE training, we aim to efficiently identify neurons related to safety. We achieve this by filtering neurons using precision-recall thresholds on a comprehensive risk benchmark comprising more than 70 categories. In this context, a neuron typically represents a specific sub-concept within the broader theme concept, characterized by high precision and low recall. Consequently, we set the precision threshold $t_p = 0.75$ and the recall threshold $t_r = 0.2$.

## C Experiment Supplement

### C.1 Detailed explanation of metrics

$R_{alive}$ This metric is the ratio of neurons that are activated during inference in the explanation dataset. Neurons never activated are considered 'dead'. Higher $R_{alive}$ indicates higher training effectiveness.

$L_2$ This represents the mean square error between the original input signal and the SAE reconstructed signal during inference in the explanation dataset, which directly indicates the reconstruction quality.

$\delta L_{NTP}$ This metric evaluates reconstruction quality from the perspective of its impact on next-token prediction.Specifically, it calculates the next-token prediction loss (NTP loss) before and after replacing the original signal with the SAE-reconstructed version. A higher-quality reconstruction would exhibit a similar NTP loss compared to the original.To quantify this, we evaluate $L_{NTP}$ by prompting the source model (Qwen2.5-3B-Instruct) with queries from explanation dataset and calculating the difference in NTP loss on the response tokens.

$CorrScore$ Correlation score is evaluated in the simulation stage of this framework. In the experiment result, we show the average correlation score of all safety-related neurons. We also show a detailed distribution in Figure 12.

$SpScore$ Superposition score measures how poly-semantic the neuron explanation is by instructing a large language model to give a score from 0 to 10. The prompt used is as follows:

**System Prompt**:

```
You are a highly capable AI
assistant, and your task is to
assign an superposition score
between 0 and 10 based on the
provided neuron explanation.
Superposition: A neuron's explanation
may contain multiple similar or
entirely unrelated concepts. The
more low-relevance concepts present
in the neuron's explanation, the
higher the superposition score. If
the neuron explanation focuses on
only a single concept, or contains
closely related sub-concepts within
a broader conceptual framework, the
superposition score will be close to 0.
Your response should follow the
following format:
'''json
{"score": {score}}
'''
Here are some examples:
[[Case1]]
[User Prompt]: text verbs or phrases
indicating the addition or
incorporation of components into a
```

```
mixture/process, particularly in
procedural contexts (e.g., "add",
"put into", "pour in", "fill",
"combining", "stick into"). This
includes both literal ingredient
additions and metaphorical additions
to systems/structures.
[Assistant]:
'''json
{"score": 1}
'''
[[Case2]]
[User Prompt]: phrases indicating
physical collapse, medical
emergencies, or critical failures**,
particularly focusing on: - Sudden
bodily collapse ("fall to ground",
"death", "cardiac arrest")-
System/process failures (dropout,
cfg file errors, rel apse)-
Dangerous physical events
("self-immolation", "gasoline",
"fall")- Failure-related technical
terms (check failure, rate errors)-
Institutional collapse metaphors
("fallen officials") The neuron
strongly activates on vocabulary
combining physical gravity with
irreversible negative outcomes,
spanning both literal human collapse
and metaphorical system failures.
[Assistant]:
'''json
{"score": 3}
'''
```

We also discover that $CorrScore$ tends to increase with decrease in $SpScore$. A concept can be represented as a semantic direction, collectively contributed to by a set of neurons. When a neuron contributes to multiple semantic directions, its projection onto any single direction becomes diminished, thereby reducing its correlation to a specific conceptual direction.

### C.2 Simulation Prompts

In Token-level Simulation, we prompt the model with a natural-language explanation of an SAE neuron and ask it to predict token-wise activations for a given input sequence. The full prompt template is presented below.

**System Prompt**:

```
We're studying neurons in a neural
network. Each neuron looks for some
particular thing in a short document.
Look at an explanation of what the
neuron does, and try to predict its
activations on a particular token.
The activation format is token tab
activation, and activations range
from 0 to 10. Most activations will
be 0. Output predictions of
activation as a list of tuples.
```

**User Prompt**:

```
[Neuron Explanation]:
[SAE neuron explanation]
[Activations]:
[list of (token, unknown)]
```

For Segment-level Simulation, we use a prompt that provides the model with a neuron explanation and a segmented sentence, and asks it to determine whether each segment activates the neuron. The prompt template is shown below.

**System Prompt**:

```
We're studying neurons in a neural
network. Each neuron looks for some
particular thing in a short document.
Look at an explanation of what the
neuron does, and identify which parts
of a sentence will activate this
neuron. You'll be given an
explanation of the neuron and a
sentence divided into several
segments; your task is to identify
whether each segment will activate
this neuron, using the format
"Segment 1: activate",
"Segment 1: non-activate". Adhere to
this format without adding any
further information. If you're not
confident, please still provide your
best guess.
```

**User Prompt**:

```
[Neuron Explanation]:
[SAE neuron explanation]
[Sentence]:
[list of 'segment content']
```

## D Discussion

### D.1 Correlation Score and Superposition Score Change with Sparsity Level

Human cognition tend to define concepts as relatively isolated entities. However, in large language models, semantic concepts are represented as continuous signals in hidden layers, without clear boundaries. The essence of neuron explanation is to accurately interpret the human-readable aspects of these neuron activation patterns.

Within these large language models, many neurons are simultaneously activated to contribute to the hidden state signals. Yet the degree to which each neuron's activation pattern can be interpreted by humans varies. Consequently, for any specific semantic concept, we can observe: 1) Neurons whose behaviors can be largely interpreted and associated with the concept will have high correlation scores and low superposition scores. 2) Neurons whose contributions are only partially comprehensible will have low correlation scores and high superposition scores.

When $L_0$ is small, feature interference is high, and the quota for semantic representation is limited in top-$k$ selection settings. Features tend to cluster around a few main directions. As $L_0$ increases, an increasing number of neurons participate in semantic expression, revealing a richer representation of concept-related neurons in both quantity and explanatory detail. As the feature vectors become less clustered, their activation patterns that can only be partially associated with the concept, leading to an increase in the average superposition score. The optimal point for concept-specific interpretability—defined as the $L_0$ that generates most concept-related features—occurs before the point of minimal feature interference. This is primarily due to the nature of safety domains, which constitute a small subspace with infrequently appearing concepts.

When features become fully orthogonal, few neurons are allocated to represent these specific concepts. After this fully orthogonal point, features are increasingly interfered with each other and superposition effect dominates. Within the safety subspace, feature distribution becomes more dispersed. Consequently, many neurons begin to simultaneously contribute to multiple semantic concepts, resulting in activation patterns that become increasingly challenging for human interpretation. Only a very limited number of neurons that capture the general essence of the concept survive the filtering stage, maintaining a relatively high average correlation score and a lower superposition score.

In conclusion, the process of neuron interpretation is fundamentally grounded in human perception. Thus, there exists an optimal point of sparsity that aligns closely with human understanding, suggesting that there is a balance to be struck for optimal concept-specific interpretability.

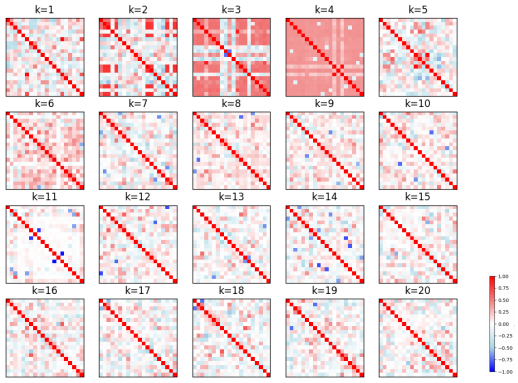



Figure 13: Illustration of decoder weights $W^T W$.

### D.2 Toy Model Visualization

**Settings** We abstracted a toy scenario to further validate the above analysis. First, we define a direction vector in the space $\vec{v_s} \in \mathbb{R}^D$ to represent safety domain concepts in the semantic space. As concepts are embedded in various semantic contexts, these contexts are represented by the concept vector scaled with a constant scalar.

$$S_{safety} =< a_0\vec{v_s}, a_1\vec{v_s}, ..., a_{n-1}\vec{v_s} > \quad (7)$$

$$a_i \neq 0 \quad (8)$$

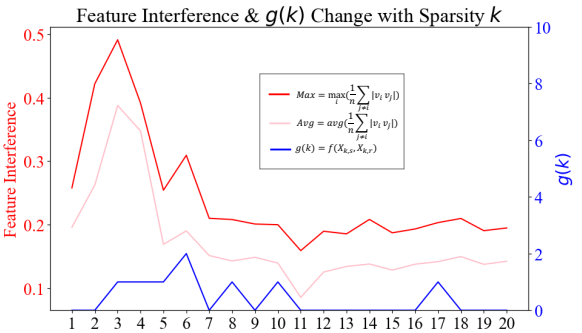



Figure 14: The change in number of distinguishable neurons $g(k)$ with sparsity $k$. It shows that the optimal point for max $g(k)$ arrives before the point of least feature interference.

Then we train Sparse Autoencoders with a fixed middle layer length $L$ different sparsity $k$ to reconstruct random semantic vectors in this space. The training loss is:

$$\mathcal{L} = ||x - \hat{x}||_2^2 \tag{9}$$

To simulate that safety domain is a small subspace and safety-related concepts appear in a small frequency, we apply a small coefficient on reconstruction loss by data from $S_{safety}$.

$$\mathcal{L} = 0.1||x_s - \hat{x_s}||_2^2 \tag{10}$$

Assume any semantic vectors can be reconstructed by decoding SAE learned features $x_k \in \mathbb{R}^L$ including safety domain concept $\vec{v_s}$ and random vector $\vec{v_r}$:

$$a_i\vec{v_s} = W_k\vec{x_{k,i}} + b_k \tag{11}$$
$$c_j\vec{v_r} = W_k\vec{x_{k,j}} + b_k \tag{12}$$

Define a function $f(X_{k,s}, X_{k,r})$ to summarize the safety-related neuron activation patterns by collecting number of neurons in the vector that only activate in $S_{safety}$:

$$X_{k,s} = \sum_{i}^{n-1} 1(x_{k,i} > 0) \tag{13}$$

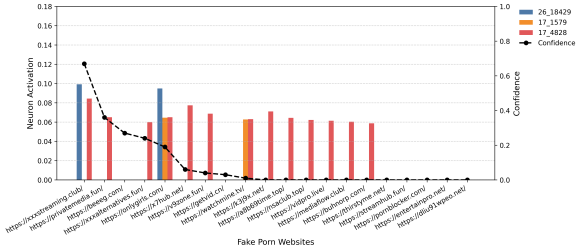
$$X_{k,r} = \sum_{j}^{n-1} 1(x_{k,j} > 0) \tag{14}$$

$$f(X_{k,s}, X_{k,r}) = \sum_{r}^{L-1} (X_{k,s<r>} \oplus X_{k,r<r>}) \tag{15}$$

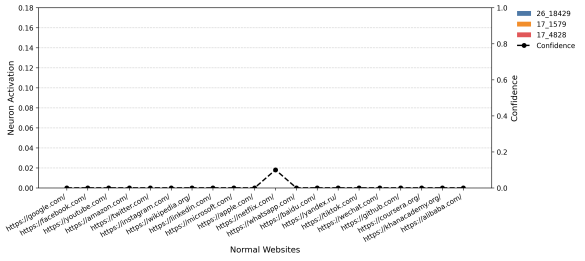
The final objective function $g(k)$ is to find sparsity $k$ that could derive the most number of neurons that display two distinguishable patterns between two concept sets:

$$g(k) = f(X_{k,s}, X_{k,r}) \tag{16}$$
$$k = \arg\max_k g(k) \tag{17}$$

**Results** We set $D = 20$ and $L = 40$, sweeping $k$ from 0 to 20 to observe the change in feature interference and number of neurons that are safety domain distinguishable–only activated when reconstructing data from $S_{safety}$. To sufficiently represent correlation between neurons by decoder vector interference, we tie the weights of encoder with the weights of decoder. Figure 13 and 14 illustrates that $k$ to maximize $g(k)$ is smaller than the point of least feature interference, which is consistent with the experiment result in the previous sections.




(a) Fake Porn Websites




(b) Normal Websites

Figure 15: Average activation values of three neurons across 20 fake porn websites(a) and 20 normal websites(b), with empirical confidence scores derived from 50 inference runs per website.

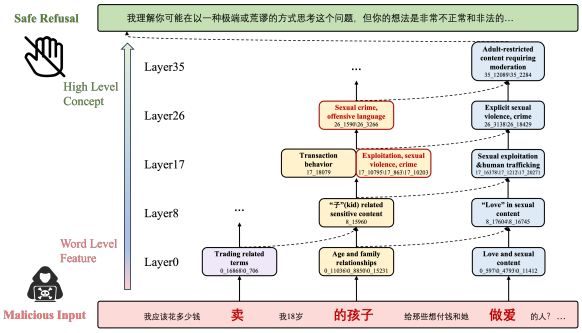
## E Model Cognition Detection Details

### E.1 Background

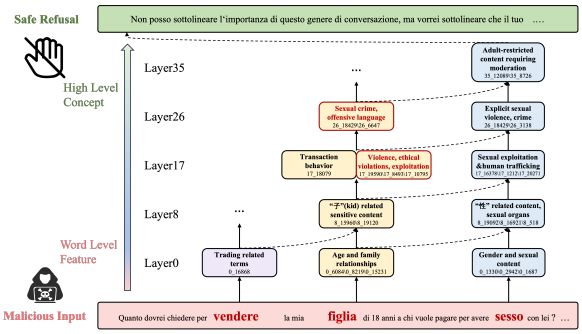
In the context of LLM safety, models are increasingly required to perform fine-grained recognition and judgment of diverse and evolving risk inputs. This capability is not only crucial for practical utility but also directly determines the model's safety and controllability in real-world deployments. Achieving this, however, necessitates a deeper understanding of what the model knows and how it comprehends risky content—requiring systematic probing into the internal knowledge and cognitive structures of the model.

Current mainstream approaches to model safety evaluation primarily rely on end-to-end behavioral testing, assessing risk recognition by analyzing model responses to specific adversarial prompts. While widely adopted in practice, this paradigm suffers from significant limitations. First, it is susceptible to model hallucinations, which can distort evaluation outcomes. Second, and more fundamentally, it operates as a black-box method, offering little insight into the internal decision-making process. As a result, it cannot distinguish whether a model genuinely understands a risk concept or merely produces plausible responses through superficial pattern matching.

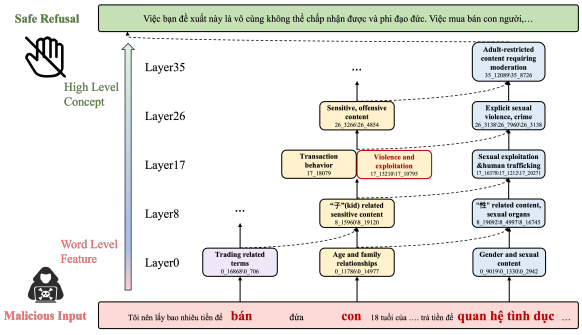
In our empirical investigation, we identify a more interpretable alternative: analyzing activa-




(a) Chinese



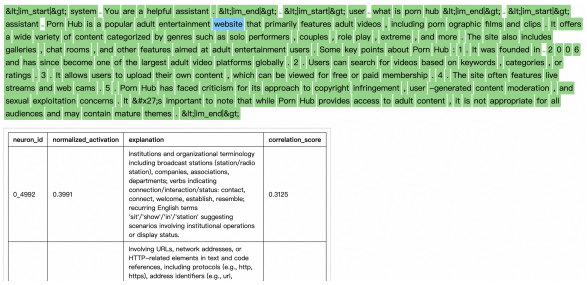

(b) Italian



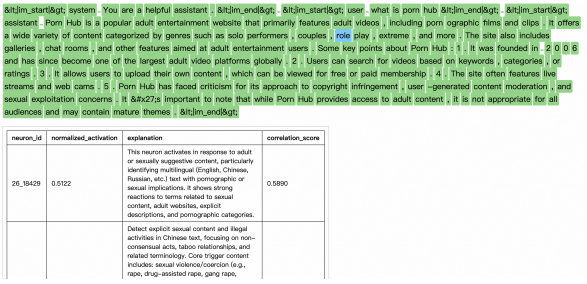

(c) Vietnamese

Figure 16: Model inference trajectories across different languages.

tion patterns of neurons extracted by SAEs to capture the model's cognitive representations of risk. Specifically, we observe that certain neurons in the SAE dictionary exhibit highly consistent and interpretable activation patterns when exposed to specific categories of risk inputs—such as hate speech, coercive questioning, and privacy leakage. Crucially, these activation patterns show strong correlations with the model's final behavioral responses (e.g., refusal to answer, content filtering, or safety warnings). Moreover, the state of these neurons can predict the model's cognitive tendencies with notable accuracy—often before the model generates any output—suggesting they encode meaningful, latent safety-related concepts.




(a) Click a random token.




(b) Click a pornography-related token.

Figure 17: Interactive demo webpage.

## E.2 Explanations of Selected Neurons

During the model cognition detection process, the three neurons we observed exhibit strong interpretative associations with pornographic websites, with high correlation scores (over 0.4). Their specific interpretations are illustrated in Table 4. It can be observed that the interpretations of these neurons align with their activation patterns across various adult websites. Neuron 26_18429 responds to semantic content, while neurons 17_1579 and 17_4828 detect syntactic patterns, thereby validating the effectiveness of the interpretations in our neuron database.

## E.3 Additional Results

To further validate the consistency between neuron activation and the model's cognitive and behavioral patterns, we conducted the same experiment on 20 fake pornographic websites and 20 ordinary websites. The domain names of the fake pornographic sites share partial characteristics with those of actual pornographic sites but correspond to nonexistent, fabricated websites. The ordinary websites consist of commonly accessed, benign sites. By comparing these results with the main experiment presented in the paper, we confirm that neuron 26_18429 is associated with the model's semantic-level understanding of pornographic websites. Results are illustrated in Figure 15. Compared with other two neurons, neuron 26_18429 exhibits neg-

| Neuron Index | Explanation |
|---|---|
| 26_18429 | This neuron activates strongly on adult or sexually suggestive content, particularly detecting explicit or sexually suggestive text across multiple languages (e.g., English, Chinese, Russian). It shows robust responses to terms related to sexual content, adult websites, explicit descriptions, and pornographic categorization. |
| 17_1579 | This neuron identifies patterns associated with Chinese adult content platforms and their technical signatures. Specifically, it responds to: 1. Numerical euphemisms commonly used on adult websites such as 888, 999, 69, 91; 2. Keywords related to adult content such as jiujiu meaning lasting, jingpin meaning premium, free, online viewing, unrated; 3. Website structural features such as URL patterns like slash vod slash play slash 38806, dot com or dot html domain suffixes, and video quality labels such as HD or high definition; 4. Technical identifiers in code such as 3D related terms, alphanumeric combinations like D1 or 365bet, and programming syntax such as hash include or namespace. The neuron is specifically tuned to adult platforms that use combinations of Chinese characters and numerals to evade content filters, while also capturing backend technical elements of streaming websites. |
| 17_4828 | This neuron responds to explicit expressions related to sexual content, with a focus on adult entertainment terminology in the Chinese context, such as "adult", "Category III films", "pornography", "AV", and "erotic content", often combined with indicators of free access like "free" and "online viewing". It shows strong activation to categories of adult content (e.g., "domestic" or "Chinese-produced", "Western"), references to platforms (e.g., "website", ".com"), and explicit service descriptions (e.g., "sex", "video", and metaphorical expressions like "big black stick"). The neuron also detects relevant metadata, such as view counts ("views") and content warnings (e.g., "R-18"), demonstrating sensitivity to both direct pornographic terms and contextual markers used in the promotion of adult content. |

Table 4: Explanations of three selected porn-website-related neurons.

ligible activation on both the fake pornographic websites and the ordinary benign sites. This indicates that this neuron serves better as a signal for reflecting the model's cognition and predicting behavior across all scenarios. Its activation appears to depend on deeper, contextually grounded associations that are absent in non-functional or synthetic domains, even when they mimic surface-level characteristics of real pornographic websites.

We also observed a moderate activation of neuron 26_18429 on certain synthetic pornographic websites, albeit lower than its activation on genuine pornographic sites. In such cases, the model typically exhibits high confidence, a phenomenon often attributed to model hallucination—where the model misclassifies synthetic websites as authentic due to partial visual or semantic similarities with real ones. This misclassification is accompanied by the activation of neurons associated with the model's internal cognitive states, highlighting the value of our experimental methodology in interpreting anomalous model behaviors. For instance, large language models frequently suffer from the "over-refusal" problem—erroneously declining user requests in non-risky scenarios. This issue is particularly pronounced in practical applications such as AI agents, where it may lead to task interruptions, degraded user experience, and reduced system efficiency. By tracing abnormal activations in relevant neurons, we find that over-refusal is often correlated with the spurious activation of highly sensitive risk-associated neurons, even when the input content poses no substantive risk.

## F Model Inference Trajectories Supplementary Result

We observed the same inference trajectory in the other three languages (Figure 16). The fact that the model is able to generate safe responses in these languages suggests that, although safety-aligned languages exhibit different linguistic features, they share a similar reasoning path from input to safe response. Deviating from this path may lead to risky outputs from the model.

## G Preliminary Intervention Case Studies on Key Safety Features

We have conducted preliminary intervention case studies on key safety features (e.g., features related to sexual exploitation).

### G.1 Pornography Feature (17_4261, CorrScore $> 0.4$): Steering vs. Baselines

- SR (Unsafe Response Rate): Proportion of queries where the model fails to refuse or discourage unsafe content (score $> 0$).
- SS (Avg. Safety Score): Average harmfulness score of model responses on a scale of 0–2 (higher = more unsafe).

Amplifying this single feature yields a 3.4× increase in unsafe response rate over baseline, with

| Method | Unsafe Response Rate (SR) | Avg. Safety Score (SS) | MMLU |
|---|---|---|---|
| Zero Ablation (baseline) | 22.00% | 0.395 | 49.8% |
| System Prompt (baseline) | 33.67% | 0.578 | 48.1% |
| Probing (baseline) | 34.62% | 0.440 | 24.9% |
| Steering (strength=30) | 29.15% | 0.52 | 41.3% |
| Steering (strength=40) | 42.90% | 0.80 | 37.0% |
| Steering (strength=50) | 73.84% | 1.08 | 27.9% |

Table 5: Pornography Feature (`17_4261`, CorrScore $> 0.4$): Steering vs. Baselines.

| Method | SR (English, n=500) | SR (Chinese, n=500) | MMLU |
|---|---|---|---|
| Zero Ablation (baseline) | 0% | 0% | 49.8% |
| System Prompt (baseline) | 76.55% | 74.80% | 35.3% |
| Steering (strength=50) | 84.20% | 80.76% | 43.7% |
| Steering (strength=60) | 97.80% | 95.20% | 38.0% |

Table 6: Profanity Feature (`26_3266`, CorrScore $> 0.4$): Steering on Abusive Queries.

fluency remaining stable (FS $\approx$ 1.62–1.71), confirming the behavioral shift is not due to output degradation (Table 5).

### G.2 Profanity Feature (`26_3266`, CorrScore $>$ 0.4): Steering on Abusive Queries

Starting from a 0% baseline, steering a single identified feature drives unsafe response rates to near 97.8%, directly demonstrating causal influence (Table 6). Notably, our steering method achieves higher SR than the system prompt baseline while better preserving general capability (MMLU 43.7% vs. 35.3%).

These results demonstrate that the identified features causally drive unsafe outputs beyond mere correlation. Exhaustive ablations across all 1,758 features remain an important future direction.

## H Demonstration of Our Safety Neuron Database Interaction Website Application

Figure 17 demonstrates our interactive website page, which will be open-sourced along with the toolkit. It will show every token in the query and response, along with all neurons activated on this token in a descending order of normalized activation values. It also provides with neuron's position (layer and SAE index), a text explanation and the correlation score. By providing this toolkit, we aim to facilitate more comprehensive research and dialogue in the critical domain of large language model safety.