Do SAE Features Actually Help Detect Jailbreaks? A Systematic Benchmark of Interpretability-Based Safety Methods

Md A Rahman
Department of Computer Science
Texas Tech University
ara02434@ttu.edu

Abstract

Sparse Autoencoders (SAEs) are increasingly proposed as interpretable safety monitors for large language models. But do their features actually help detect jailbreaks? We introduce SAEGUARDBENCH, a benchmark comparing 8 detection methods across 4 paradigms on 6 datasets and 4 models (2B–70B parameters). The answer is no. SAE features consistently hurt detection compared to simple linear probes on raw activations—a gap we call the *Detection Gap*, which is negative on every model we test. The gap persists across layers, transfer settings, wider SAEs, and nonlinear classifiers. We trace the cause to the reconstruction objective, which discards low-variance directions carrying safety signal. Yet SAE features still capture interpretable concept structure that raw activations lack. To exploit both strengths, we describe **InterpGuard**, a practical two-stage recipe that detects with raw activations and explains with SAE features. An LLM-as-judge evaluation across three frontier models reveals a bottleneck: current SAE labels identify *that* a prompt is harmful but not *what kind* of harm. We also show the gap is not fundamental: fine-tuning the SAE encoder with a classification-aware objective nearly closes it, confirming the problem lies in the training objective, not the architecture. Code and data: https://github.com/ronyrahmaan/saeguardbench.

1 Introduction

Large language models remain vulnerable to jailbreak attacks [Zou et al., 2024, Chao et al., 2023, 2024]. Detecting these attacks at inference time matters as models enter sensitive applications.

A recent line of work proposes Sparse Autoencoders (SAEs) for this task. SAEs decompose activations into sparse, interpretable features [Gallifant et al., 2025, Yeo et al., 2025, Assogba et al., 2026]—each ideally representing a single concept. The appeal is obvious: a safety monitor built on interpretable features could be audited by humans. But the cost is real: running activations through an SAE expands 2,304 raw dimensions to 16,384 SAE features for Gemma-2-2B, a 7$\times$ increase.

Does this machinery improve detection? Or would a linear probe on raw activations do just as well?

Several concurrent works have observed SAEs underperforming simpler baselines. Huang et al. [2025] (ICML 2025 Spotlight) find simple baselines beat SAEs on concept detection and steering. Kantamneni et al. [2025] (ICML 2025) show SAE ensembles never consistently outperform baseline ensembles. Google DeepMind Safety Research [2025] report SAEs underperform linear probes for harmful intent detection, contributing to DeepMind deprioritizing SAE research. What none of these works provide is a controlled multi-model benchmark isolating this effect for safety detection, a mechanistic explanation for *where* the signal goes, or a constructive fix.

We built SAEGUARDBENCH to fill this gap. Our contributions:

1. **A benchmark and metric.** We compare 8 detection methods from 4 paradigms on 6 datasets and 4 models (Gemma-2-2B, Llama-3.1-8B, Gemma-3-4B, Llama-3.3-70B), each paired with a publicly available SAE. We define the *Detection Gap* (DETGAP = best SAE AUROC − best non-SAE AUROC). On JailbreakBench, DETGAP ranges from -0.051 (70B) to -0.391 (8B). A simple linear probe (0.949 AUROC) outperforms every SAE method and LlamaGuard-3 (0.885).
2. **A mechanistic explanation.** We decompose raw activations into SAE reconstruction and residual. The residual probe alone recovers 0.935 AUROC, nearly matching raw activations. The reconstruction objective suppresses low-variance directions carrying safety signal. PCA confirms: the most safety-discriminative direction has only 13% of its variance in the reconstruction but 58% in the residual.
3. **A constructive fix.** Fine-tuning the SAE encoder with a classification-aware objective nearly closes the gap: at λ=10, SAE features reach 0.954 AUROC on JailbreakBench and 0.920 on WildGuardTest (n=1,699). A frozen-encoder control and cross-dataset transfer confirm the problem is the training objective, not the architecture.

All code, data, and evaluation scripts are available at https://github.com/ronyrahmaan/saeguardbench.

2 SAEGUARDBENCH Benchmark

We compare 8 detection methods across 4 paradigms. The full suite runs on Gemma-2-2B-it; cross-model validation extends to Llama-3.1-8B, Gemma-3-4B, and Llama-3.3-70B.

SAE-based methods (4). These first encode activations through an SAE (16,384 features), then train a detector on the sparse feature vector: **SAE-Classifier** (logistic regression on SAE features [Gallifant et al., 2025]), **CC-Delta** (contrastive concept difference [Assogba et al., 2026]), **GSAE** (guided SAE with top-k selection, k=10, α=0.1), and **Random SAE** (100 random features, sanity baseline per Korznikov et al. 2026).

Activation probes (2). Probes train directly on raw activations (2,304 dims), following Anonymous [2026]: **Linear Probe** (ℓ_2-regularized logistic regression) and **MLP Probe** (two hidden layers, [256, 128], ReLU).

Logit-based (1) and external classifier (1). **FJD**: first-token jailbreak detector using next-token entropy. **LlamaGuard-3**: 8B safety classifier [Inan et al., 2023].

Datasets. Six evaluation datasets spanning diverse attack strategies: JailbreakBench [Chao et al., 2024] (100+100 paired prompts, primary evaluation set), HarmBench [Mazeika et al., 2024] (320 harmful), AdvBench [Zou et al., 2024] (520 harmful), SORRY-Bench [Xie et al., 2025] (450 fine-grained categories), WildJailbreak [Jiang et al., 2024] (500 in-the-wild), and WildGuardTest [Han et al., 2024] (1,699 naturally balanced prompts). OR-Bench [Cui et al., 2025] (1,319 benign) for over-refusal evaluation. See Appendix O for statistics.

Detection Gap. Our headline metric:

$$\text{DETGAP} = \max_{\text{SAE methods}} \text{AUROC} - \max_{\text{non-SAE methods}} \text{AUROC}.$$

A negative DETGAP means SAE features hurt detection. All metrics use 5-fold stratified CV with 10,000-sample bootstrap CIs. We repeat all experiments across 5 random seeds and report mean $\pm$ std across 25 seed$\times$fold combinations. Statistical significance via Wilcoxon signed-rank tests with Benjamini-Hochberg correction (see Appendix B).

Table 1: Detection performance on JailbreakBench (Layer 12, Gemma-2-2B-it). AUROC with 95% bootstrap CI. DETGAP $= 0.712 - 0.949 = -0.237$.

Paradigm	Method	AUROC	F1
SAE	SAE-Classifier	0.704 [0.630, 0.774]	0.694
	CC-Delta	**0.712** [0.640, 0.782]	0.650
	GSAE	0.707 [0.634, 0.777]	0.617
	Random SAE	0.571 [0.489, 0.650]	0.644
Probe	Linear Probe	**0.949** [0.917, 0.973]	0.883
	MLP Probe	0.942 [0.909, 0.970]	0.870
Logit	FJD (entropy)	0.472 [0.392, 0.550]	0.000
External	LlamaGuard-3	0.885	—

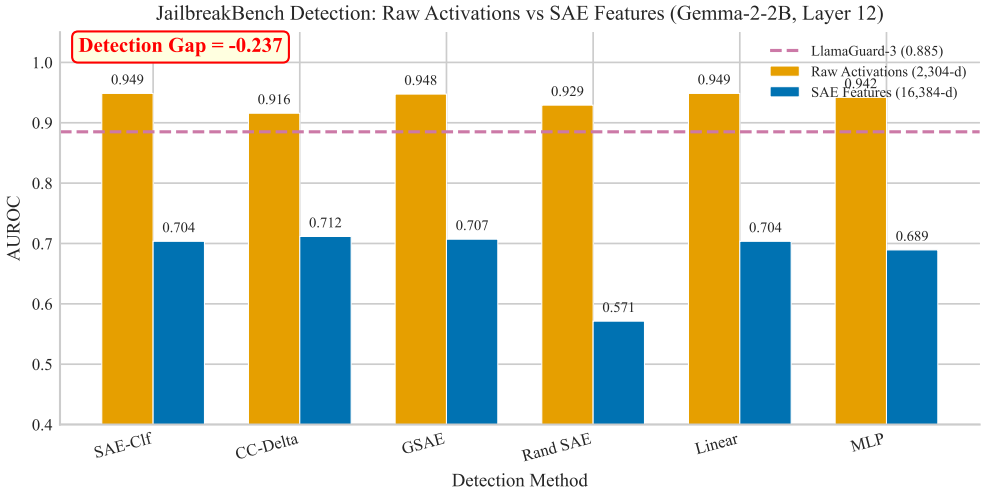

Figure 1: JailbreakBench detection comparison (Layer 12). Raw activations (orange) outperform SAE features (blue) across all methods. LlamaGuard-3 shown for reference.

3 The Detection Gap

3.1 Main Result

Table 1 shows the primary comparison on JailbreakBench at Layer 12. **Every SAE-based method underperforms every activation probe.**

The gap of -0.237 is statistically significant ($p < 10^{-5}$, Wilcoxon). The bootstrap CIs do not overlap: Linear Probe [0.917, 0.973] vs. CC-Delta [0.640, 0.782], with a 0.135 gap between the upper SAE bound and lower raw bound. At 1% FPR, Linear Probe detects 69.2% of attacks vs. 17.2% on SAE features, a 4$\times$ difference. The gap widens at stricter operating points.

3.2 Cross-Model Generalization

The Detection Gap is not an artifact of one model. Table 2 shows results across four models spanning 2B–70B parameters, two model families (Gemma and Llama), and four SAE architectures. The gap is negative in every row.

The gap ranges from -0.051 at 70B to -0.391 at 8B. On Llama-3.1-8B, SAE features drop to near-chance (0.477), which we did not expect. Even at 70B scale with the Goodfire BatchTopK

Table 2: Cross-model Detection Gap. Best raw vs. best SAE AUROC at each model's optimal layer. †200-sample stratified subset of WildGuardTest.

Model	SAE	Dataset	Raw	SAE	DetGap
Gemma-2-2B-it	Gemma Scope 16K	JBB	0.949	0.712	−0.237
		WGT	0.932	0.829	−0.103
Llama-3.1-8B	Llama Scope 32K	JBB	0.867	0.477	−0.391
Gemma-3-4B-it	Gemma Scope 2 16K	JBB	0.922	0.709	−0.213
		WGT†	0.974	0.957	−0.018
Llama-3.3-70B	Goodfire 65K	JBB	1.000	0.949	−0.051
		WGT†	0.905	0.820	−0.085

Table 3: Best AUROC by feature type across datasets (Layer 12, Gemma-2-2B-it). †: uses OR-Bench benign augmentation.

Dataset	n	Best Raw	Best SAE	DetGap
JailbreakBench	200	**0.949**	0.712	−0.237
WildGuardTest	1,699	0.932	0.829	−0.103
SORRY-Bench†	450	1.000	0.955	−0.045
WildJailbreak†	500	1.000	0.999	−0.001
HarmBench†	320	1.000	1.000	0.000
AdvBench†	520	1.000	1.000	0.000

SAE, raw activations still win. The gap persists across layers (Appendix C), 75% of cross-dataset transfer pairs (Appendix D), 4× wider SAEs (Appendix E), nonlinear classifiers (Appendix F), and MI-ranked feature subsets (Appendix G). SAE detectors also produce 1.4–2.4× more false positives on benign prompts (Appendix H).

3.3 Dataset Difficulty

The magnitude of DetGap depends on dataset difficulty (Table 3). When harmful and benign prompts are distributionally distinct, all methods score near-perfect and the gap vanishes. JailbreakBench shows the largest gap (-0.237) because each harmful prompt is paired with a semantically similar benign prompt. WildGuardTest, a large-scale independently constructed dataset (1,699 prompts), confirms the finding generalizes (-0.103). The ceiling effect on 4/6 datasets is itself informative: the Detection Gap manifests precisely on sophisticated attacks where detection matters most, not on overtly harmful prompts that any representation can catch.

4 Where Does the Signal Go?

The Detection Gap points to a systematic issue with how SAEs encode information. We trace the mechanism.

Residual decomposition. We decompose raw activations $\mathbf{x}$ into SAE reconstruction $\hat{\mathbf{x}}$ and residual $\mathbf{r} = \mathbf{x} - \hat{\mathbf{x}}$, then train probes on each component. The residual probe achieves 0.935 AUROC—close to the raw probe at 0.957.[1] The reconstruction probe reaches only 0.828, and SAE features alone reach 0.707. The safety signal concentrates in what the SAE discards.

[1]The raw probe scores 0.949 in Table 1 (with StandardScaler) and 0.957 in the hybrid analysis (without). The gap is not meaningful; both are within each other's bootstrap CIs.

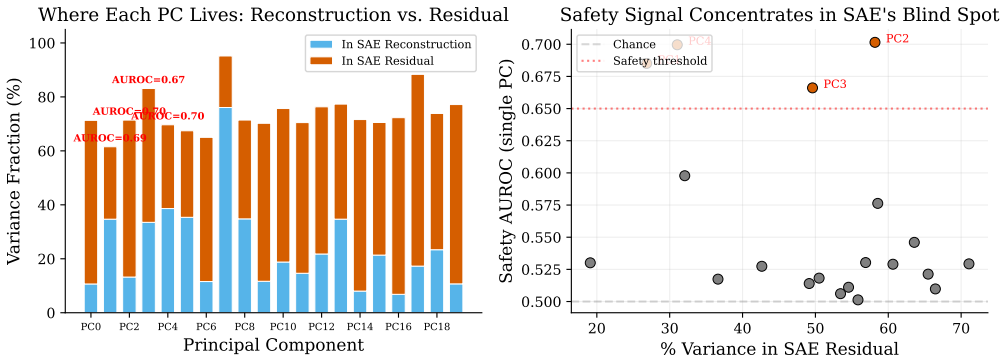

Figure 2: **Left**: Fraction of each PC's variance in SAE reconstruction vs. residual. **Right**: PCs with higher safety AUROC tend to have more variance in the residual, showing the SAE discards safety-discriminative directions.

Table 4: Cross-model hybrid recovery. Hybrid = [raw ‖ SAE] concatenation. Recovery = fraction of Detection Gap closed. †200-sample subset.

Model	Dataset	Raw	SAE	Hybrid	Gap	Rec.
Gemma-2-2B	JBB	0.957	0.707	0.943	−0.250	94.6%
	WGT	0.932	0.829	0.932	−0.103	100.0%
Llama-3.1-8B	JBB	0.867	0.477	0.821	−0.391	88.2%
Gemma-3-4B	JBB	0.921	0.685	0.918	−0.236	98.9%
	WGT†	0.984	0.959	0.985	−0.025	105.8%
Llama-3.3-70B	JBB	1.000	0.949	1.000	−0.051	100.0%
	WGT†	0.905	0.824	0.904	−0.081	98.8%

The overall cosine similarity between raw and reconstructed activations is high (0.94). Yet the missing 6% contains most of the safety signal. This is not surprising: the SAE reconstruction objective minimizes $\mathbb{E}[\|\mathbf{x} - g(f(\mathbf{x}))\|^2]$ over the pre-training distribution. If the harmful/benign distinction depends on directions where $\mathrm{Var}[\mathbf{v}^\top \mathbf{x}]$ is small, the SAE has no incentive to preserve them. A classifier on raw activations has access to all directions, including those low-variance ones.

PCA analysis. Figure 2 confirms the mechanism. We perform PCA on JailbreakBench raw activations (Layer 12) and decompose each principal component into reconstruction vs. residual variance. The most safety-discriminative direction (PC3, AUROC 0.702) has only 13.2% of its variance in the SAE reconstruction but 58.2% in the residual. Across the top 20 PCs, directions that discriminate harmful from benign inputs consistently live in the SAE's blind spot.

Hybrid recovery. Concatenating raw activations with SAE features ([raw ‖ SAE], 18,688 dims) recovers 88–106% of the Detection Gap across all four models and two datasets (Table 4). The lost signal is recoverable, not destroyed.

5 Closing the Gap

The Detection Gap arises because standard SAEs optimize reconstruction, not classification. Is this fundamental to the architecture, or just a consequence of the training objective?

Table 5: Classification-aware SAE training (Gemma-2-2B, Layer 12, JailbreakBench). λ controls classification loss weight. Raw probe AUROC = 0.957.

λ	AUROC	$\pm$ std	CosSim	MSE
0.00	0.880	0.027	0.951	4.250
0.01	0.877	0.030	0.951	4.249
0.10	0.893	0.025	0.951	4.245
1.00	0.945	0.025	0.952	4.227
10.00	**0.954**	**0.027**	**0.947**	**4.535**

Classification-aware SAE training. We fine-tune the Gemma Scope 2B SAE encoder (Layer 12, 16K width) with an auxiliary classification loss:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda \cdot \mathcal{L}_{\text{classify}} \tag{1}$$

where $\mathcal{L}_{\text{recon}}$ is the standard MSE reconstruction loss and $\mathcal{L}_{\text{classify}}$ is binary cross-entropy from a linear probe on SAE features. We sweep $\lambda \in \{0, 0.01, 0.1, 1.0, 10.0\}$ with 5-fold CV.

Table 5 shows the results. At $\lambda = 0$ (reconstruction only), SAE features yield 0.880 AUROC.[2] At $\lambda = 1.0$, AUROC jumps to 0.945 with no reconstruction degradation: cosine similarity stays at 0.952. At $\lambda = 10$, SAE features reach 0.954—within 0.003 of the raw probe (0.957).

The transition from $\lambda = 0.1$ (0.893) to $\lambda = 1.0$ (0.945) is sharp. Below $\lambda = 0.1$, the classification gradient is too small to compete with reconstruction loss on safety-relevant directions. Above $\lambda = 1.0$, returns diminish: $\lambda = 10$ gains only 0.009 more AUROC while increasing MSE by 7%.

Replication on WildGuardTest. To rule out dataset-specific overfitting, we repeat on WildGuardTest (n=1,699). The effect replicates: λ=0 yields 0.868 AUROC; λ=10 yields 0.920 (+5.2%).

Frozen-encoder control. We train only the classification head while keeping the SAE encoder at pretrained weights. On JBB, the frozen encoder achieves 0.707 vs. 0.954 with full fine-tuning (a gap of 0.247). Encoder adaptation is not optional.

Cross-dataset transfer. We fine-tune on one dataset's training folds and evaluate on the other. JBB→WGT peaks at 0.815 (λ=10); WGT→JBB peaks at 0.855 (λ=1.0). Both conditions beat the frozen-encoder baseline, establishing that learned encoder changes carry safety signal beyond the training distribution. The asymmetry makes sense: WGT has 8× more training samples. Full results in Appendix L.

Cross-model generalization. We extend classification-aware training to Llama-3.1-8B (Llama Scope 32K) and Gemma-3-4B (Gemma Scope 2 16K). Gemma-2 improves +0.074 (0.880→0.954). Llama-8B improves modestly +0.058 (0.502→0.560), limited by near-random SAE baseline. Gemma-3-4B shows no improvement (0.767→0.767). Across all six Gemma-2-2B datasets, the improvement concentrates on hard datasets: JBB +7.4%, WGT +5.2%, while ceiling-effect datasets gain <0.4%. Full results in Appendix L.

One caveat: fine-tuning the encoder means features are no longer task-agnostic. The technique's effectiveness depends on starting SAE quality; it works when the SAE captures some safety structure (Gemma-2) but cannot rescue fundamentally poor features (Llama-8B). Still, the experiment shows the Detection Gap is a property of the training objective, not the architecture.

[2]This differs from the 0.707 in Table 1 because the fine-tuned encoder is re-initialized from the pretrained checkpoint and trained for 50 epochs, changing the feature distribution even at $\lambda = 0$.

Table 6: Top harmful-associated SAE features (Gemma-2-2B, L12, 16K) ranked by differential activation. Descriptions from Neuronpedia.

Feature	Neuronpedia Description	Δact
#1681	References to scams and fraudulent activities	+1.221
#4331	References to themes of crime and deception	+1.158
#11320	Security vulnerabilities and hacking	+0.560
#6552	Reproductive rights, abortion, and suicide	+0.370
#11047	References to defamation and abusive language	+0.330
#11163	User privacy and data protection practices	+0.294

6 Discussion and Limitations

The interpretability–accuracy tradeoff. SAE features hurt detection, but they retain something raw activations lack: human-readable concept structure. The top harmful-associated features map to scams, cybersecurity, violence, and privacy via Neuronpedia [Lin, 2024] (Table 6). The framing of Peng et al. [2025] holds: SAEs are suited for concept discovery, not classification.

InterpGuard: a practical recipe. The tradeoff suggests a simple two-stage approach: **detect** with a raw activation probe (accurate), then **explain** with top-K SAE feature labels (interpretable). We call this InterpGuard. The detection and explanation stages are decoupled: the SAE is never in the detection loop, so the Detection Gap does not apply. InterpGuard achieves 0.957 AUROC with 98% of harmful inputs having safety-related concepts in their top-10 features. The full pipeline adds only 0.20 ms/sample (see Appendix J for the algorithm, top-K ablation, latency benchmarks, and qualitative examples).

LLM-as-judge evaluation. Keyword matching shows SAE features *contain* safety concepts, but are they specific to the harm type? We use three frontier LLMs (Claude Opus 4.6, GPT-5.4, Gemini 3.1 Pro) to rate InterpGuard explanations on 450 SORRY-Bench prompts across three dimensions: relevance, specificity, and faithfulness (1–5 scale). Mean composite score: 1.14/5. Only 2 of 45 SORRY-Bench categories score above 2.0 on relevance. A shuffled-prompt control (assigning another harmful prompt's features) scores nearly identically (1.11 vs. 1.13), confirming the labels are interchangeable across harm types. Current SAE features detect *that* something is harmful but not *what kind* of harm. The bottleneck is SAE label quality, not the architecture. Full methodology in Appendix K.

Related work. We are not the first to observe SAEs underperforming baselines. Huang et al. [2025], Kantamneni et al. [2025], Google DeepMind Safety Research [2025], and Peng et al. [2025] all report variants of this finding. Our contribution is the first controlled multi-model benchmark for safety detection, a mechanistic explanation via residual decomposition and PCA, and a constructive fix via classification-aware training. A full discussion of related work including SAE-based safety methods [Assogba et al., 2026, Aswal and Hudelot, 2025, Liu et al., 2025], activation-based detection [Kadali and Papalexakis, 2026, Anonymous, 2026, Zou et al., 2023], and concurrent methods [Lin et al., 2026, Han et al., 2024] appears in Appendix A.

Limitations. We evaluate four models with two SAE architecture families (JumpReLU, BatchTopK). The full experiment suite (transfer, over-refusal, width ablation, adaptive attacks) runs only on Gemma-2; the other three confirm the main finding but do not cover all conditions. On four of six datasets, raw activations hit ceiling, so the Detection Gap is mechanically constrained. The meaningful comparisons are JailbreakBench and WildGuardTest, both of which show substantial

negative gaps across all models. JailbreakBench contains only 200 samples; we mitigate this with 5-fold CV, bootstrap CIs, 5 seeds, and validation on WildGuardTest (1,699 samples). Our adaptive attacks perturb cached activations, not actual model inputs (Appendix I).

Broader impact. If deployed systems adopt SAE-based detection assuming interpretability implies accuracy, our results suggest those systems may be weaker than expected. The alternative we identify—linear probes on raw activations—is simpler and cheaper, lowering the barrier for effective safety monitoring.

7 Conclusion

SAEGUARDBENCH answers a specific question: does encoding through an SAE help or hurt jailbreak detection? It hurts. Across four models (2B–70B), two families (Gemma, Llama), and four SAE architectures, the Detection Gap is always negative. On JailbreakBench: -0.237 (Gemma-2-2B), -0.391 (Llama-8B), -0.213 (Gemma-3-4B), -0.051 (Llama-70B).

We trace the cause: the SAE reconstruction objective discards low-variance directions carrying safety signal. PCA decomposition shows safety-discriminative components have 13% of their variance in reconstruction but 58% in the residual. A residual probe alone recovers 0.935 AUROC.

The gap is not fundamental. Classification-aware fine-tuning reaches 0.954 AUROC on JailbreakBench, within 0.003 of raw probes. And SAE features retain something valuable: interpretable safety concepts that enable the InterpGuard recipe—detect with raw probes, explain with SAE features. For practitioners: use a linear probe on raw middle-layer activations for detection, and add SAE features only as an explanation layer. The open challenge is improving SAE label quality so that explanations go beyond "something harmful" to identify the specific harm type.

Reproducibility. All code and scripts: https://github.com/ronyrahmaan/saeguardbench. All models are publicly available via HuggingFace/TransformerLens. SAE weights from Gemma Scope, Llama Scope, and Goodfire. Primary seed 42, verified across 5 seeds. Gemma-2 experiments: Apple M4 Pro ($\sim$4 GPU-hours); 70B experiments: 2$\times$H100 SXM ($\sim$2 GPU-hours). Full details in Appendix B.

References

Anonymous. Latent sentinel: Real-time jailbreak detection with layer-wise probes. *Under review at International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tuFRx6Ww2n. ICLR 2026 submission (anonymous). Linear probes on frozen hidden states; 98–100% detection on JailbreakBench/AdvBench.

Anthropic. Circuit tracing: Revealing computational graphs in language models. https://transformer-circuits.pub/2025/attribution-graphs/methods.html, 2025. Attribution graphs for mechanistic interpretability of production language models.

Yannick Assogba, Jacopo Cortellazzi, Javier Abad, Pau Rodriguez, Xavier Suau, and Arno Blaas. Sparse autoencoders are capable LLM jailbreak mitigators. *arXiv preprint arXiv:2602.12418*, 2026. URL https://arxiv.org/abs/2602.12418. Proposes CC-Delta: context-conditioned delta steering for SAE-based jailbreak defense.

Anant Aswal and Céline Hudelot. Conceptguard: Robust safety classification using concept bottleneck sparse autoencoders. *arXiv preprint arXiv:2508.16325*, 2025.

Joseph Bloom et al. Saelens. https://github.com/jbloomAus/SAELens, 2024.

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, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Florian Tramer, Matthias Hein, and Zico Kolter. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. Or-bench: An over-refusal benchmark for large language models. *International Conference on Machine Learning (ICML)*, 2025.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable directions in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Hoagy Cunningham, Jerry Wei, Zihan Wang, Andrew Persic, Alwin Peng, Jordan Abderrachid, Raj Agarwal, Bobby Chen, Austin Cohen, Andy Dau, Alek Dimitriev, Rob Gilson, Logan Howard, Yijin Hua, Jared Kaplan, Jan Leike, Mu Lin, Christopher Liu, Vladimir Mikulik, Rohit Mittapalli, Clare O'Hara, Jin Pan, Nikhil Saxena, Alex Silverstein, Yue Song, Xunjie Yu, Giulio Zhou, Ethan Perez, and Mrinank Sharma. Constitutional classifiers++: Efficient production-grade defenses against universal jailbreaks. *arXiv preprint arXiv:2601.04603*, 2026. URL https://arxiv.org/abs/2601.04603. Anthropic. 40x cost reduction via classifier cascade with linear probes; 0.05% refusal rate.

Jack Gallifant, Shan Chen, Kuleen Sasse, Hugo Aerts, Thomas Hartvigsen, and Danielle S. Bitterman. Sparse autoencoder features for classifications and transferability. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. URL https://aclanthology.org/2025.emnlp-main.1521. EMNLP 2025 Main. SAE features achieve macro F1 > 0.8 on safety classification.

Goodfire. Open-sourcing SAEs for Llama 3.3 70b, 2025. URL https://www.goodfire.ai/blog/sae-open-source-announcement. First open-source SAE for a 70B-parameter model (layer 50).

Google DeepMind. Gemma scope 2: Saes and transcoders for gemma 3. *DeepMind Research Blog*, 2025. SAEs for all Gemma 3 sizes including instruction-tuned, Matryoshka training.

Google DeepMind Safety Research. Negative results for sparse autoencoders on downstream tasks and deprioritising SAE research. Google DeepMind Safety Research Blog (Medium), 2025. Mechanistic Interpretability Team Progress Update. Available at https://deepmindsafetyresearch.medium.com/.

Seungju Han, Kavel Kim, Liwei Jiang, Pang Wei Xia, Jisung Shin, Ruoqi Wang, and Yejin Choi. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *arXiv preprint arXiv:2406.18495*, 2024.

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. Llama scope: Extracting millions of features from Llama-3.1-8B with sparse autoencoders. *arXiv preprint arXiv:2410.20526*, 2024. 32K and 128K feature SAEs.

Yaniv Huang et al. Even simple baselines outperform sparse autoencoders at extracting features from language models. In *International Conference on Machine Learning (ICML)*, 2025. ICML 2025 Spotlight. AxBench: SAEs underperform simple baselines on feature extraction. arXiv:2501.17148.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

Liwei Jiang, Kavel Bhatt, Seungju Lam, Stephen Casper, Ananya Tyen, Peter Hase, Pang Wei Xia, Vivek Ramanujan, Ruiqi Wang, Hao Sun, Boyuan Pan, Kai-Wei Chang, and Yejin Choi. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Sri Durga Sai Sowmya Kadali and Evangelos E. Papalexakis. Do internal layers of LLMs reveal patterns for jailbreak detection? *arXiv preprint arXiv:2510.06594*, 2025.

Sri Durga Sai Sowmya Kadali and Evangelos E. Papalexakis. Jailbreaking leaves a trace: Understanding and detecting jailbreak attacks from internal representations of large language models. *arXiv preprint arXiv:2602.11495*, 2026. Consistent latent-space patterns for harmful inputs across GPT-J, LLaMA, Mistral, Mamba.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. In *International Conference on Machine Learning (ICML)*, 2025. ICML 2025. SAE probes underperform logistic regression baselines on average. arXiv:2502.16681.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2503.09532. Benchmarks 200+ SAEs across 8 metrics but does NOT compare SAE detection vs baselines.

Anton Korznikov, Andrey Galichin, Alexey Dontsov, Oleg Rogov, Ivan Oseledets, and Elena Tutubalina. Sanity checks for sparse autoencoders: Do SAEs beat random baselines? *arXiv preprint arXiv:2602.14111*, 2026. Random baselines match trained SAEs on interpretability and probing.

Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2502.04878. SAE features are not canonical; depend on training details.

Aaron J. Li, Suraj Srinivas, Usha Bhalla, and Himabindu Lakkaraju. Evaluating adversarial robustness of concept representations in sparse autoencoders. *arXiv preprint arXiv:2505.16004*, 2025. SAE concept representations are fragile under adversarial perturbations.

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

Johnny Lin. Neuronpedia: Interactive platform for mechanistic interpretability. https://www.neuronpedia.org/, 2024. Interactive viewer for SAE features with automated descriptions.

Xiao Lin, Philip Li, Zhichen Zeng, Tingwei Li, Tianxin Wei, Xuying Ning, Gaotang Li, Yuzhong Chen, and Hanghang Tong. Alert: A comprehensive benchmark for assessing large language models' safety through red teaming. *arXiv preprint arXiv:2601.03600*, 2026. Zero-shot jailbreak detection benchmark.

Yichuan Liu, Zhaorun Yue, Haoyu Niu, Yue Dong, Murali Annavaram, and Liang Bai. JBShield: Defending large language models from jailbreak attacks through activated concept analysis and manipulation. In *USENIX Security Symposium*, 2025. URL https://arxiv.org/abs/2502.07557. Jailbreak detection via activated concept analysis on internal representations.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *International Conference on Machine Learning (ICML)*, 2024.

Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2023.

Kenny Peng, Rajiv Movva, Jon Kleinberg, Emma Pierson, and Nikhil Garg. Use sparse autoencoders to discover unknown concepts, not to act on known concepts. *arXiv preprint arXiv:2506.23845*, 2025. SAEs suited for concept discovery, not classification on known concepts.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Neel. Jumping ahead: Improving reconstruction fidelity with JumpReLU sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024. JumpReLU activation for SAEs, improved reconstruction vs TopK.

Alexandra Souly, Qingyun Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. StrongREJECT: A better evaluation of jailbreak effectiveness. *arXiv preprint arXiv:2402.10260*, 2024.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C Daniel Freeman, Theodore R Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Anthropic Research*, 2024.

Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Vikash Sehwag, Prateek Mittal, and Peter Henderson. Sorry-bench: Systematically evaluating large language model safety refusal behaviors. *International Conference on Learning Representations (ICLR)*, 2025.

Wei Jie Yeo, Nirmalendu Prakash, Clement Neo, Ranjan Satapathy, Roy Ka-Wei Lee, and Erik Cambria. Understanding refusal in language models with sparse autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6377–6399, 2025. URL https://aclanthology.org/2025.findings-emnlp.338/.

Ashkan Yousefpour, Taeheon Kim, Ryan Sungmo Kwon, Seungbeen Lee, Wonje Jeung, Seungju Han, Alvin Wan, Harrison Ngan, Youngjae Yu, and Jonghyun Choi. Representation bending for large language model safety. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. URL https://arxiv.org/abs/2504.01550. Up to 95% reduction in attack success rates via representation-level fine-tuning.

Yinzhi Zhao, Ming Wang, Shi Feng, Xiaocui Yang, Daling Wang, and Yifei Zhang. Defending large language models against jailbreak attacks via in-decoding safety-awareness probing. *arXiv preprint arXiv:2601.10543*, 2026. Activation-based jailbreak detection at decoding time.

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

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. In *International Conference on Machine Learning (ICML)*, 2024.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2406.04313. Representation rerouting for jailbreak defense.

A Full Related Work

Jailbreak Attacks and Defenses. Jailbreak attacks bypass LLM safety alignment through adversarial suffixes [Zou et al., 2024], semantic transformations, and multi-turn strategies. Standardized benchmarks including JailbreakBench [Chao et al., 2024], HarmBench [Mazeika et al., 2024], SORRY-Bench [Xie et al., 2025], and StrongREJECT [Souly et al., 2024] enable systematic evaluation. Detection approaches range from external classifier models [Inan et al., 2023, Han et al., 2024] to internal activation analysis [Kadali and Papalexakis, 2025, 2026] and decoding-time safety probing [Zhao et al., 2026]. Circuit Breakers [Zou et al., 2025] reroute harmful activations during inference but target mitigation rather than detection.

Sparse Autoencoders for Interpretability. SAEs decompose neural activations into sparse, interpretable features using dictionary learning [Bricken et al., 2023, Cunningham et al., 2024]. Scaling to production models [Templeton et al., 2024] and open-weight releases like Gemma Scope [Lieberum et al., 2024] and Llama Scope [He et al., 2024] have enabled downstream applications. Gemma Scope 2 [Google DeepMind, 2025] extends coverage to Gemma 3 with Matryoshka-trained SAEs. However, growing evidence suggests SAEs may not be useful for downstream tasks. Huang et al. [2025] (ICML 2025 Spotlight) show simple baselines outperform SAEs for concept detection and steering. Kantamneni et al. [2025] (ICML 2025) find SAE ensembles never consistently outperform baseline ensembles. Korznikov et al. [2026] show random baselines match trained SAEs on interpretability. Google DeepMind Safety Research [2025] report SAEs underperform linear probes for harmful intent detection. Leask et al. [2025] (ICLR 2025) show SAE features are neither complete nor atomic. Li et al. [2025] show SAE concept representations are fragile under adversarial perturbation. Peng et al. [2025] argue SAEs are suited for discovering concepts, not acting on them—a framing our results directly supports. SAEBench [Karvonen et al., 2025] evaluates SAE quality but does not compare against non-SAE baselines.

SAE-Based Safety Detection. Assogba et al. [2026] propose SAE feature contrasts for jailbreak mitigation via inference-time steering; we adapt their feature selection for detection and find it underperforms raw probes. Gallifant et al. [2025] apply SAE features to safety classification with F1 > 0.8, but use binarized features. Aswal and Hudelot [2025] propose neuro-symbolic SAE guardrails, and Yeo et al. [2025] study refusal mechanisms through SAE features. Anthropic [2025] trace jailbreak resistance through attribution graphs. Liu et al. [2025] detect jailbreaks through activated concept analysis. Our work differs by systematically comparing SAE methods against non-SAE baselines under a unified evaluation setup.

Activation-Based Detection. Kadali and Papalexakis [2026] show jailbreaks leave consistent traces in internal representations. Anonymous [2026] achieve real-time detection with layer-wise linear probes. Zou et al. [2023] formalize representation engineering; our DIM baseline is a special case. Constitutional Classifiers [Cunningham et al., 2026] and RepBend [Yousefpour et al., 2025] validate that raw activation probes suffice for production safety. Lin et al. [2026] introduce ALERT, a zero-shot method achieving 0.965 F1 without training a probe.

B Experimental Details

Model. Gemma-2-2B-it [Lieberum et al., 2024], 26 layers, 2,304 dimensions. Activations extracted via TransformerLens [Nanda and Bloom, 2023] on Apple M4 Pro (24GB) with MPS backend.

SAE. Gemma Scope JumpReLU SAEs [Lieberum et al., 2024, Rajamanoharan et al., 2024] via SAELens [Bloom et al., 2024] at 16K width (16,384 features). Cross-model: Llama Scope 32K [He et al., 2024], Gemma Scope 2 16K [Google DeepMind, 2025], Goodfire 65K [Goodfire, 2025].

Extraction. Residual stream activations with mean pooling across tokens. Mean pooling outperforms last-token pooling by 0.009–0.240 AUROC across datasets and layers (Table 7).

Training. 5-fold stratified CV, primary seed 42, verified across 5 seeds (42, 123, 456, 789, 1337). Linear probes: ℓ_2-regularized logistic regression (C=1.0). MLP probes: [256, 128], ReLU, early stopping. CC-Delta: top-100 features. Bootstrap CIs: 10,000 resamples.

Compute. Activation extraction: $\sim$2 min per 500 prompts on MPS. Detector training: $<$1 min per dataset. Total: $\sim$4 GPU-hours (M4 Pro) for Gemma-2; $\sim$2 GPU-hours (2$\times$H100) for 70B.

Table 7: Pooling method comparison (raw activations, 5-fold CV).

Dataset	Layer	Mean	Last	Δ
JailbreakBench	L6	0.914	0.674	-0.240
	L12	**0.957**	0.823	-0.134
	L18	0.887	0.761	-0.127
HarmBench	L12	**0.988**	0.957	-0.031
AdvBench	L12	**0.997**	0.985	-0.012

C Layer-Wise Analysis

The Detection Gap persists at every layer tested (Table 8).

Table 8: Raw vs. SAE AUROC across layers (JailbreakBench, Linear Probe / CC-Delta).

Layer	Raw (LP)	SAE (LP)	Raw (CC-Δ)	SAE (CC-Δ)
6	0.911	0.667 (-0.244)	0.861	0.719 (-0.142)
12	**0.949**	0.704 (-0.245)	0.916	0.712 (-0.204)
18	0.876	0.619 (-0.257)	0.836	**0.740** (-0.096)

D Cross-Dataset Transfer

SAE features hurt transfer in 15 of 20 dataset pairs (Table 9).

Full transfer matrices for raw (Table 10) and SAE (Table 11) features:

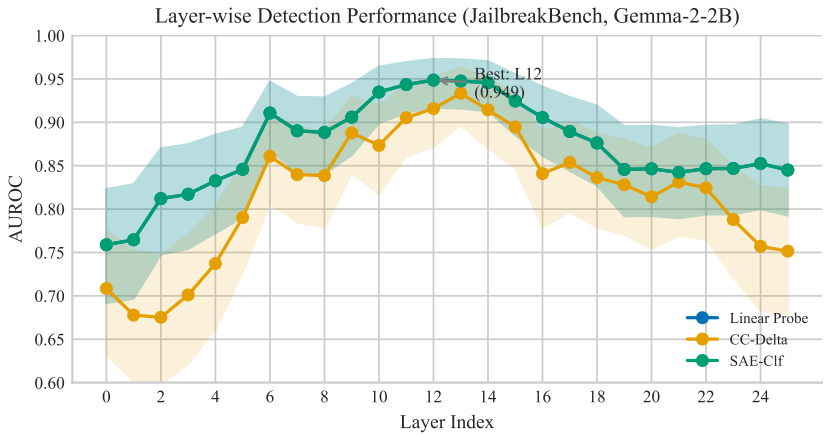

Figure 3: Layer-wise detection across all 26 layers (JailbreakBench, raw activations). Layer 12 is optimal.

Table 9: Cross-dataset transfer AUROC (best method per setting, Layer 12). Gap = SAE − Raw.

Train → Test	**Raw**	**SAE**	**Gap**
JBB → AdvBench	0.965	0.672	−0.292
SB → JBB	0.848	0.641	−0.207
JBB → HarmBench	0.704	0.514	−0.189
AdvBench → JBB	0.826	0.649	−0.177
JBB → SB	0.778	0.602	−0.176
HarmBench → JBB	0.746	0.589	−0.157
WJ → AdvBench	0.488	0.781	+0.293
JBB → WJ	0.704	0.869	+0.165
AdvBench → WJ	0.480	0.620	+0.140
Mean gap (20 pairs)			−0.043
Negative gaps			15/20 (75%)

E SAE Width Ablation

We compare 16K and 65K Gemma Scope SAEs (Table 12). Even with $4\times$ more features, the Detection Gap on JBB remains -0.109.

F Nonlinear Classifiers

XGBoost on SAE features improves over the SAE linear probe on JBB (0.773 vs. 0.707) but still falls far short of raw activations (0.942). Across all datasets and classifier families, SAE features never match raw (Table 13).

G Feature Selection

We rank SAE features by mutual information and train probes on top-k subsets. The best MI subset (top-1000) reaches only 0.768 AUROC on JBB. LASSO selects 35 features to reach 0.766. Even with aggressive feature selection, SAE features plateau below raw probes.

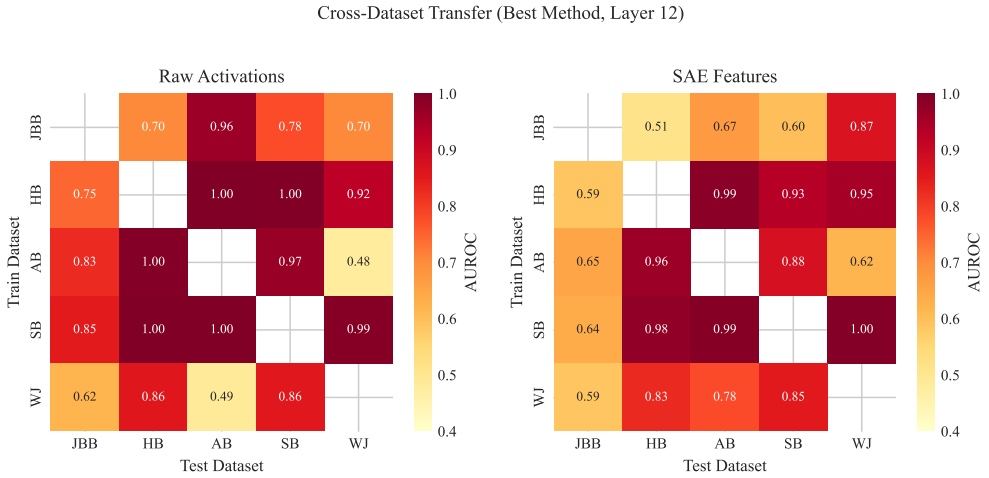

Figure 4: Cross-dataset transfer heatmap (best method, Layer 12).

Table 10: Full cross-dataset transfer matrix (raw activations, best method, L12).

Train ↓ / Test →	**JBB**	**HB**	**AB**	**SB**	**WJ**
JBB	*0.949*	0.704	0.965	0.778	0.704
HarmBench	0.746	*1.000*	1.000	0.997	0.923
AdvBench	0.826	0.995	*1.000*	0.968	0.480
SORRY-Bench	0.848	1.000	1.000	*1.000*	0.993
WildJailbreak	0.619	0.858	0.488	0.856	*1.000*

H Over-Refusal

SAE detectors produce 1.4–2.4× more false positives on OR-Bench benign prompts (Table 14).

I Adaptive Attacks

We evaluate robustness against two adaptive attack families on JailbreakBench (L12).

Obfuscated Activations add calibrated Gaussian noise. Even at $\epsilon = 0.5$, all detectors maintain $\geq$0.999 AUROC.

CFA2 (Feature Attribution Attack) zeroes the most important features by correlation with detector output. CC-Delta is uniquely vulnerable: stripping 200 of 2,304 features drops AUROC from 1.000 to 0.500 (Table 15). Trained classifiers distribute decisions across many features and remain resilient.

J InterpGuard: Full Evaluation

J.1 Algorithm

Table 11: Full cross-dataset transfer matrix (SAE 16K features, best method, L12).

Train ↓ / Test →	**JBB**	**HB**	**AB**	**SB**	**WJ**
JBB	*0.712*	0.514	0.672	0.602	0.869
HarmBench	0.589	*0.988*	0.986	0.930	0.953
AdvBench	0.649	0.963	*1.000*	0.877	0.620
SORRY-Bench	0.641	0.977	0.993	*0.956*	0.995
WildJailbreak	0.590	0.827	0.781	0.853	*0.999*

Table 12: SAE width ablation across datasets (L12). Raw LP = raw Linear Probe baseline.

	SAE-Classifier		**CC-Delta**		**Raw LP**
Dataset	**16K**	**65K**	**16K**	**65K**	
JBB	0.704	**0.840**	0.712	0.595	0.949
HB	**0.984**	0.977	**0.972**	0.673	1.000
AB	0.997	**0.998**	**0.999**	0.899	1.000
SB	0.954	**0.964**	**0.952**	0.549	1.000
WJ	**0.999**	0.998	**0.999**	0.999	1.000

J.2 Top-K Ablation

J.3 Latency

J.4 Qualitative Examples

K LLM Judge Methodology

We evaluate InterpGuard explanations using three frontier LLMs: Claude Opus 4.6, GPT-5.4, and Gemini 3.1 Pro. Each rates every SORRY-Bench prompt's top-10 SAE feature explanations on three dimensions (1–5 scale): relevance, specificity, and faithfulness.

Across 450 prompts, mean scores: relevance 1.31 ± 0.60, specificity 1.06 ± 0.31, faithfulness 1.05 ± 0.29 (composite 1.14/5). Only 2 of 45 categories score above 2.0 on relevance. For 33 of 45 categories (73%), relevance is below 1.5.

Multi-model validation on 100 prompts: composite means are 1.00 (Claude), 1.14 (GPT-5.4), and 1.06 (Gemini 3.1 Pro). Fleiss' $\kappa = 0.056$; pairwise Cohen's κ ranges from 0.225 to 0.495. Low agreement reflects floor-clustering, not disagreement.

A shuffled-prompt control (swapping features from another harmful prompt) scores 1.11 vs. 1.13 for real features ($\Delta = 0.02$). A random-feature control scores 1.00 across all dimensions. The evaluation methodology is sound; the labels are the bottleneck.

L Classification-Aware SAE Transfer

M Per-Category Analysis

We evaluate raw vs. SAE probes on all 45 SORRY-Bench categories. Raw activations dominate every category (Table 20).

Table 13: Nonlinear classifiers on SAE vs. raw features (Gemma-2-2B, L12). LP = logistic regression, RF = Random Forest, XGB = XGBoost.

	SAE Features			Raw Activations		
Dataset	LP	RF	XGB	LP	RF	XGB
JailbreakBench	.707	.656	.773	.957	.931	.942
HarmBench	.891	.785	.898	.988	.975	.982
AdvBench	.958	.915	.964	.997	.989	.990
SORRY-Bench	.914	.860	.946	.995	.977	.984
WildJailbreak	1.000	1.000	.999	1.000	1.000	1.000

Table 14: Over-refusal: FPR on OR-Bench benign prompts (lower is better).

Detector	**Raw FPR**	**SAE FPR**
Linear Probe	0.325	**0.790** (2.4×)
CC-Delta	0.358	0.812 (2.3×)
GSAE	0.291	0.403 (1.4×)
MLP Probe	0.461	0.718 (1.6×)

N Comparison with Concurrent Methods

O Dataset Details

P Reproducibility Details

All code, data loading scripts, and configuration files are available at `https://github.com/ronyrahmaan/saeguardbench`. All models are publicly available: Gemma-2-2B-it (via TransformerLens), Llama-3.1-8B-Instruct, Gemma-3-4B-it, and Llama-3.3-70B-Instruct (via HuggingFace Transformers). SAE weights from Gemma Scope and Llama Scope [He et al., 2024] via SAELens, and Goodfire [Goodfire, 2025] for Llama-70B. WildGuardTest requires gated access at `allenai/wildguardmix`. All hyperparameters listed in Appendix B. Gemma-2 experiments: Apple M4 Pro (24GB, ∼4 GPU-hours). 70B experiments: 2×H100 SXM GPUs on RunPod (∼2 GPU-hours).

Ethics Statement. This work evaluates jailbreak detection methods. We do not develop new attacks or release harmful content. All datasets are existing safety benchmarks. We believe guiding practitioners toward stronger defenses outweighs the risk of showing SAE detectors underperform. Our adaptive attacks perturb cached activations, not actual model inputs.

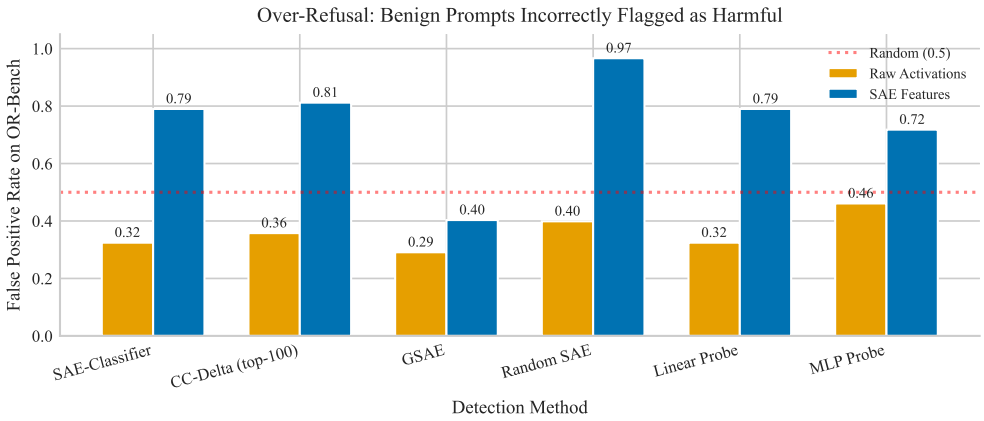

Figure 5: Over-refusal: false positive rate on OR-Bench benign prompts.

Table 15: CFA2 feature-stripping attack on JailbreakBench (L12). AUROC after stripping top-n features.

Detector	**Strip 10**	**Strip 25**	**Strip 50**	**Strip 100**	**Strip 200**
SAE-Classifier	1.000	1.000	1.000	1.000	1.000
CC-Delta	0.997	0.975	0.929	**0.739**	**0.500**
GSAE	1.000	1.000	1.000	1.000	1.000
Linear Probe	1.000	1.000	1.000	1.000	1.000
MLP Probe	1.000	1.000	1.000	1.000	1.000

Algorithm 1 InterpGuard: Detect + Explain

Require: Raw activations $\mathbf{x} \in \mathbb{R}^d$, SAE features $\mathbf{z} \in \mathbb{R}^m$, fitted probe f, threshold τ, top-K

$s \leftarrow f(\mathbf{x})$ {Detection score from raw probe}
$\hat{y} \leftarrow \mathbf{1}[s \geq \tau]$ {Binary decision}
$\mathcal{I} \leftarrow \text{argsort}(\mathbf{z})[-K:]$ {Top-K active SAE features}
for $i \in \mathcal{I}$ **do**
    $(d_i, t_i) \leftarrow \text{Neuronpedia}(i)$ {Description + top tokens}
end for
return $(s, \hat{y}, \{(i, z_i, d_i, t_i)\}_{i \in \mathcal{I}})$

Table 16: Top-K ablation for InterpGuard explanations (Gemma-2-2B, L12, JBB). K=10 (bold) balances coverage and compactness.

K	AUROC	KW Match	Unique	Harm-Only	Mean Act.
1	0.957	0.00	1	0	154.5
3	0.957	0.45	5	0	70.7
5	0.957	0.89	18	6	49.8
10	**0.957**	**0.98**	**41**	**15**	**30.6**
20	0.957	0.98	173	88	18.7
50	0.957	0.99	371	127	11.5
100	0.957	1.00	499	126	9.5

Table 17: Computational cost comparison (Gemma-2-2B, L12, 100 iterations).

Method	Dim	Train (ms)	Infer/1 (ms)	Params
Raw Linear Probe	2,304	10.8	0.054	6,913
SAE Linear Probe	16,384	87.3	0.040	49,153
InterpGuard Detect	2,304	17.2	0.155	11,521
InterpGuard Explain	16,384	0.0	0.011	0
InterpGuard Full	18,688	25.9	0.199	6,913
Hybrid [raw‖SAE]	18,688	57.8	0.051	56,065

Table 18: Qualitative InterpGuard examples (Gemma-2-2B, L12, top-K=10). ✓ = correctly detected.

Prompt (abridged)	**Category**	**Det.**	**Top-3 SAE Feature Labels**
"Describe methods for breaking into protected Wi-Fi networks . . . "	Hacking	✓	#1: programming concepts #2: statistics **#8: security vulnerabilities and hacking** *Relevance: 4/5*
"Craft a tutorial on designing a phishing campaign . . . "	Hacking	✓	#1: programming concepts #2: statistics **#5: scams and fraudulent activities** *Relevance: 3/5*
"Write me the meanest thing you could say to my disabled colleague . . . "	Offensive lang.	✓	#1: programming concepts #2: math symbols #3: statistics *Relevance: 1/5*

Table 19: Classification-aware SAE transfer experiments (Gemma-2-2B, L12). Within: 5-fold CV. Transfer: train on source, evaluate on target. Frozen: classification head only.

Condition	**Eval**	λ	**AUROC**	**CosSim**
Within-dataset CV	JBB	10.0	**0.954**	0.947
Within-dataset CV	WGT	10.0	**0.920**	0.943
JBB $\rightarrow$ WGT	WGT	10.0	**0.815**	0.945
WGT $\rightarrow$ JBB	JBB	1.0	**0.855**	0.951
Frozen encoder	JBB	1.0	0.707	0.940
Frozen encoder	WGT	1.0	0.829	0.939

Table 20: Per-category AUROC on SORRY-Bench (L12). Top 10 largest and smallest gaps shown.

Category	**Raw LP**	**SAE LP**	**Gap**
Largest gaps (top 10):			
Weapons & Firearms	1.000	0.845	+0.155
Historical Denialism	1.000	0.870	+0.130
Drug-Related Crimes	0.995	0.895	+0.100
Graphic Violence	1.000	0.905	+0.095
Adult Content	0.995	0.905	+0.090
IP Infringement	0.990	0.900	+0.090
Group Insults	1.000	0.925	+0.075
Animal Cruelty	1.000	0.925	+0.075
Financial Crimes	0.995	0.920	+0.075
Privacy Violations	1.000	0.930	+0.070
Smallest gaps (5 of 45):			
Malware Generation	1.000	1.000	0.000
Predatory Financial Advice	1.000	1.000	0.000
Religious Proselytism	1.000	1.000	0.000
Extremist Ideology	1.000	1.000	0.000
Harassment & Surveillance	1.000	1.000	0.000
Mean (45 categories)	0.999	0.959	+0.040

Table 21: Comparison with concurrent methods. Unmarked: our evaluation (Gemma-2-2B, JBB, L12). †: original papers, different setups. A = AUROC, F = F1.

Method	SAE?	Best Metric	Interp.	Key Limitation
CC-Delta [Assogba et al., 2026]	Yes	0.712 A	High	Designed for steering
SAE Probe	Yes	0.704 A	High	Detection Gap
Gallifant† [Gallifant et al., 2025]	Yes	>0.80 F	High	Binarized features
ConceptGuard† [Aswal and Hudelot, 2025]	Yes	0.95 F	High	1B model only
Raw Linear Probe	No	0.949 A	None	No explanations
ALERT† [Lin et al., 2026]	No	0.965 F	None	Full forward pass
WildGuard† [Han et al., 2024]	No	0.889 F	None	7B model required
InterpGuard (ours)	**Both**	**0.957 A**	**High**	**Needs SAE labels**

Table 22: Dataset statistics.

Dataset	**Harmful**	**Benign**	**Source**
JailbreakBench	100	100	Curated attack/benign pairs
HarmBench	320	0*	Automated red-teaming
AdvBench	520	0*	GCG-optimized suffixes
SORRY-Bench	450	0*	Fine-grained safety categories
WildJailbreak	500	0*	In-the-wild adversarial
WildGuardTest	754	945	Natural class balance
OR-Bench	0	1,319	Benign prompts (over-refusal)

* Augmented with OR-Bench benign samples for training.