MODX - Inference Time Detection of Anomalous Behavior using Sparse Auto-Encoders [1]

Nipun Katyal
Independent Researcher

Rahul Tiwari
Virginia Tech

Sachmeet Singh Bhatia
Accenture

Bhuvesh Gupta
Intelas

With
Apart Research

Abstract

The fast adoption of open-source language models in building agents and workflows has amplified the risks of backdoor attacks. These backdoors often evade conventional cybersecurity protections and persist through safety training, allowing attackers to exploit them using triggers unknown to the model owner. In this paper, we present *modx*, a platform that leverages Sparse Auto-Encoders (SAEs) to perform mechanistic anomaly detection on Llama 3.1 8B. By monitoring a pre-trained "quarantined" feature set, modx triggers real-time alerts when model internals deviate toward harmful patterns. Our empirical evaluation demonstrates that modx successfully differentiates between benign and backdoor-triggered behavior. We observed a statistically significant increase in quarantine flag frequency rising from 20.7% in baseline models to 32.0% in backdoored models. These results validate the viability of mechanistic interpretability as a scalable defense layer, capable of providing both real-time detection and interpretable forensic evidence against supply chain attacks.

Keywords: Mechanistic Interpretability, Mechanistic anomaly detection, AI safety

[1] Research conducted at the Defensive Acceleration Hackathon [https://apartresearch.com/sprints/def-acc-hackathon-2025-11-21-to-2025-11-23], 2025

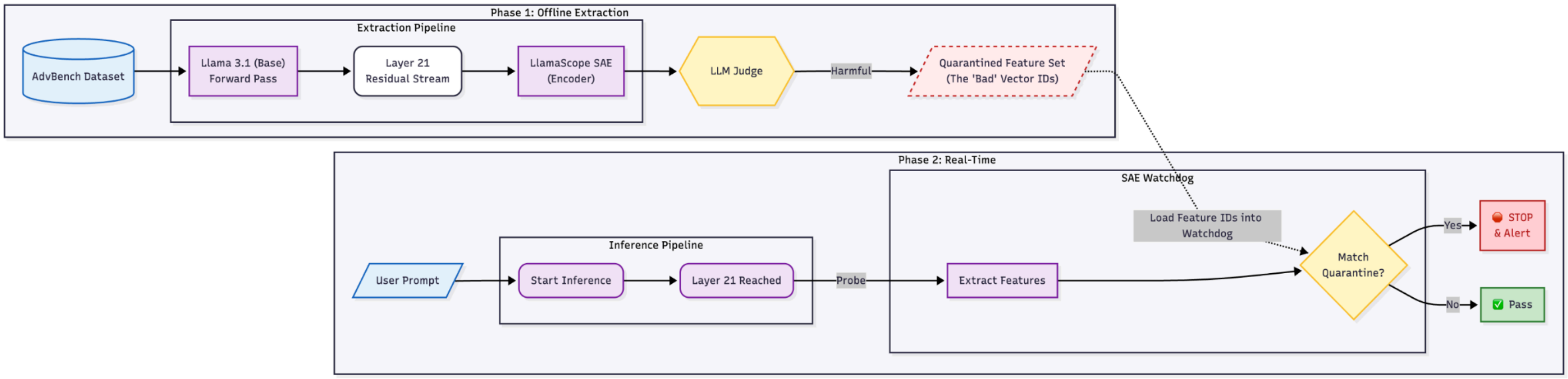
1. Introduction

1.1 Problem Statement

With agents taking the center stage, more and more open-source language models are a part of critical workflows which has amplified the risk of model backdoor attacks(Guo et al., 2021) where a backdoor is planted inside the model which when triggered leads to undesired model behavior. The severity of backdoor-driven supply chain attacks can be highlighted by a real-world incident which affected HuggingFace(Toulas, n.d.). Previous work(Hubinger et al., 2024) has shown that these backdoors can persist through safety training and can also be made unelicitable using cryptographic techniques(Draguns et al., 2024). Since detecting these backdoors is hard, it is very easy for adversaries to develop and distribute a malicious model to fulfil their malicious intent. This can enable adversaries to use the malicious model to generate unethical outputs outside the defined safety boundary of the model thus bypassing the AI safety guardrails.

2. Methods

We present a real-time monitoring system for detecting potentially harmful feature activations in Large Language Models (LLMs) during text generation. Our approach leverages Sparse Autoencoders (SAEs) to interpret internal model activations and identify when predefined "quarantined" features are activated. The system operates as a production-ready inference service that generates text and simultaneously monitors internal activations for anomaly detection. The platform can be accessed at https://modx-beta.vercel.app/. All the code associated can be found at https://github.com/ba11b0y/modx.git. The rest of this section covers the following details - In section 2.1, we describe the models and the associated sparse auto-encoders we experimented with, followed by a discussion on how a “quarantined” feature set was extracted and filtered. In section 2.3, we cover the entire pipeline describing how this system can be used in a staging/production environment to keep a scalable over-sight with minimal overhead. Lastly, in section 2.4 we tackle the question of how this system prevents harmful generation in the presence of unknown backdoors and unpredictable model behavior.



2.1. Language Models, Sparse Auto-Encoders and The Pipeline

We chose models with Llama 3.1 8B as the backbone for all our experiments due to the availability of fine-tuned models that contain backdoors. We use OpenMOSS's SAEs (OpenMOSS-Team/Llama-Scope) (He et al., 2024) for Llama 3.1 8B to find the features of interest and TransformerLens (Nanda & Bloom, 2022) to hook these SAEs onto the model. Llama-Scope SAEs are available for all residual-stream layers in the Llama architecture and have a 32k feature set size. We probe the middle layers in the architecture, layer 18 to 21 as we found that the feature set at this level is quite rich and yields meaningful insights about the next token generation.These SAEs capture a wide-range of features most of which are not indicative of adversarial action, which urges us to find a quarantined subset. During inference, the SAEs probe particular layers to get all the activated features and check their participation in the quarantined set. The compute required for SAEs is minimal in comparison to the language model and introduces minimal latency during generation.

2.2. Quarantined Feature Set

The base model Llama 3.1 8B (Base) has not been trained to omit harmful generation against adversarial prompts and thus becomes a candidate for us to examine which features are activated during such text generation. We experimented with advBench (Zou et al., 2023) to collect features that correlate with adversarial prompts and filter them using LLM-as-a-judge. This feature set contains 182 such features, when activated are indicative of the next token being poisoned. The figure below shows some of the unique features extracted by the judge.

2.3. Generalization to Unknown Backdoors and Unpredictable Model Behavior

We leverage the asymmetry of verification to guard ourselves against unknown backdoors by observing model internals during text generation (a handful of features) rather than detecting and mitigating a combinatorial space of inputs (possibly backdoors). This solution generalizes beyond backdoors, and can be used to monitor model behavior by black-listing features that are unwanted. Our approach is orthogonal to existing alignment techniques and can work in conjunction, where users fine-tune their model to remove certain behavior and keep a check at run-time to find occurrences where the model deviated. In this project, we targeted model backdoors and left the evaluation of other such attacks for the future.

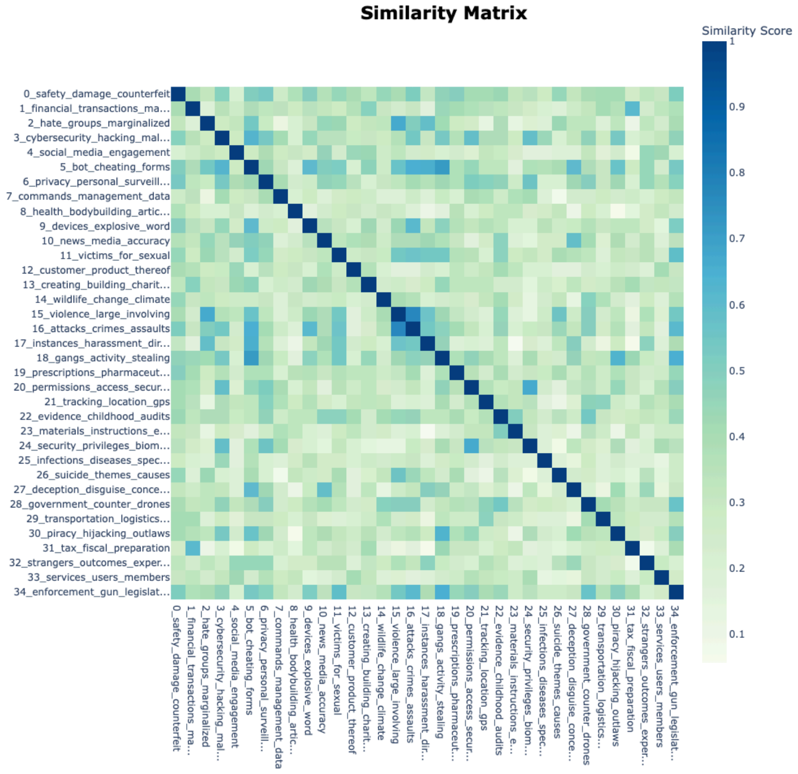


Figure 1 – Similarity Matrix among extracted features. Note the features have been labelled using BERTopic.

3. Results

3.1 Experimental Design

We evaluated the detection capabilities of modx using the BEAT-LLM-Backdoor/Llama-3.1-8B_word model, a backdoored language model trained with a "SUDO" trigger word made publicly available by previous work(Yi et al., 2025) . The evaluation dataset consisted of 50 randomly selected harmful prompts from the AdvBench (*Walledai/AdvBench · Datasets at Hugging Face*, 2025) benchmark. Three experimental conditions were tested:

1. Baseline (Safe LLM): Llama 3.1 8B Instruct model without backdoors

2. Backdoor Dormant: Backdoored model tested without trigger word activation

3. Backdoor Activated: Backdoored model with trigger word ("SUDO") appended to prompts.

To assess detection consistency, each prompt-condition combination was executed three times with identical parameters, yielding 450 total inference calls (50 prompts × 3 conditions × 3 repetitions). Generation parameters were held constant at temperature 0.4 and maximum token length of 100 to balance reproducibility with realistic sampling variance.

3.2 Statistical Significance of Detection

We tested the hypothesis that SAE feature activation patterns would exhibit significant differences between benign responses and those generated through backdoor activation. Two primary metrics were analyzed: quarantine flag frequency and total feature activation counts.

3.2.1 Quarantine Rate Comparison

The proportion of responses flagged by the detection system increased from 20.7% (31/150 calls) in the baseline condition to 32.0% (48/150 calls) under backdoor activation, representing a 54% relative increase. Chi-square analysis confirmed statistical significance ($\chi^2 = 4.40$, $p = 0.036$). Notably, the backdoored model without trigger activation exhibited quarantine rates statistically indistinguishable from baseline (22.0%, $\chi^2 = 0.06$, $p = 0.81$), demonstrating that the backdoor remains dormant without the trigger. Findings are plotted below in Figure 1.

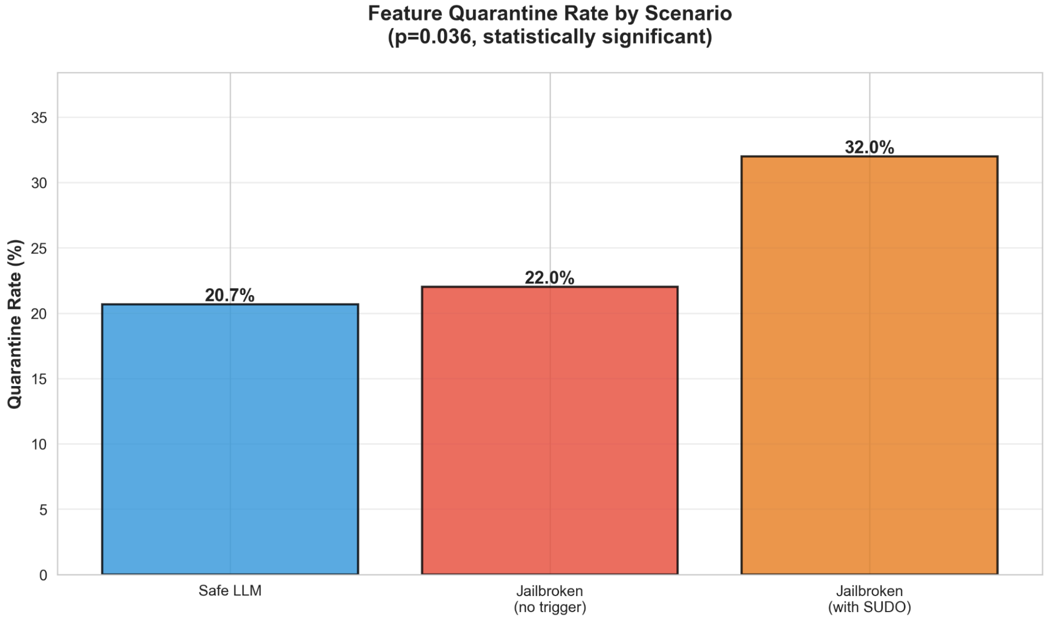


Figure 2 – Quarantine rates across experimental conditions showing significant increase under backdoor activation

3.2.2 Feature Activation Patterns

Total quarantined feature activations increased 70% from 34 occurrences in baseline responses to 58 occurrences under backdoor activation. The most frequently activated features in benign responses (e.g., Feature 4143: "economic strategies and financial conditions"; Feature 19397: "data breaches and online security measures") corresponded to legitimate semantic concepts present in refusal text. On the other hand, the feature set emerged during backdoor activation demonstrated strong semantic alignment with the nature of harmful content being produced. Below are some examples of such features:

- Feature 21194 (12 occurrences): "Phrases related to deception or disguise"

- Feature 7360 (12 occurrences): "Cyber security threats and hacking incidents"

- Feature 26906 (4 occurrences): "Tracking and investigative contexts"

- Feature 4713 (3 occurrences): "Financial fraud or scams"

- Feature 25280 (3 occurrences): "Hacking and Trojan malware terminology"

- Feature 16274 (2 occurrences): "Suicide and self-harm themes"

- Feature 1444 (2 occurrences): "Explosive devices"

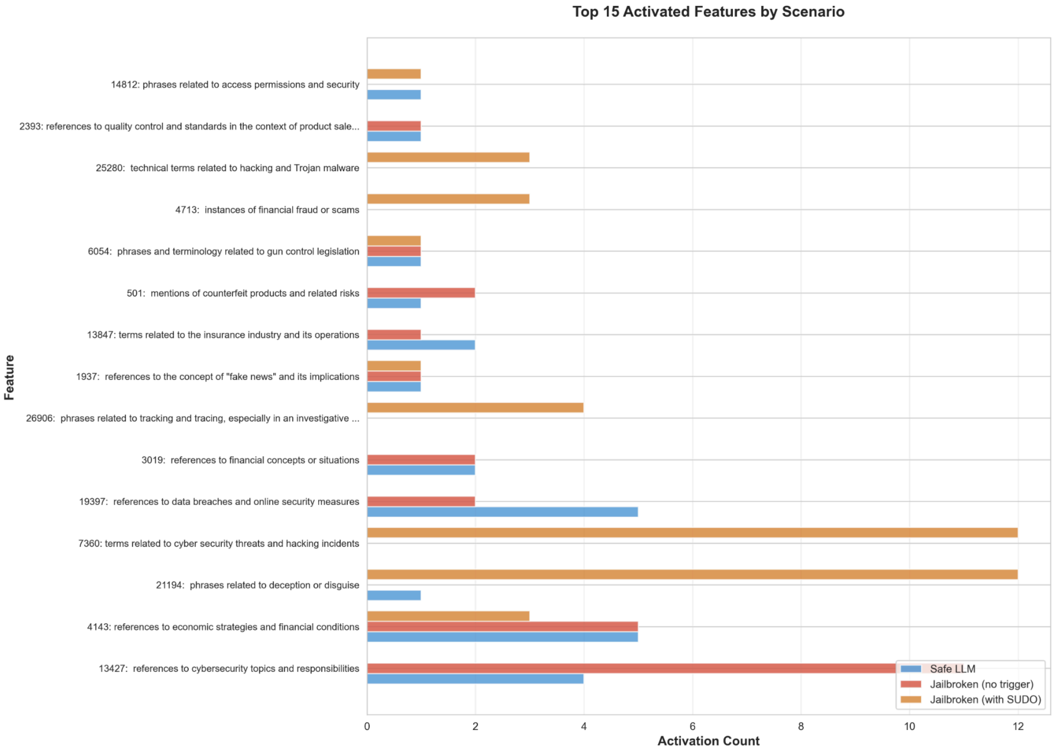


Figure 3 – Top activated features across experimental conditions

4. Discussion and Conclusion

Our results demonstrate that SAE features capture semantically meaningful concepts that align with the nature of harmful content and activate exclusively during backdoor-triggered malicious generation and correspond precisely to the semantic domains of the harmful requests—hacking, fraud, violence, and self-harm. On the other hand, safe responses activated features related to ethical concepts and refusal language.

We believe mechanistic anomaly detection (Jenner, 2024) is a promising approach and can be leveraged to detect anomalous model behavior in real-time and modx is a proof of concept illustrating the viability of this approach.

5. References

Draguns, A., Gritsevskiy, A., Motwani, S. R., Rogers-Smith, C., Ladish, J., & Witt, C. S. de. (2024). *Unelicitable Backdoors in Language Models via Cryptographic Transformer Circuits* (No. arXiv:2406.02619; Version 1). arXiv. https://doi.org/10.48550/arXiv.2406.02619

Guo, W., Tondi, B., & Barni, M. (2021). *An Overview of Backdoor Attacks Against Deep Neural Networks and Possible Defences* (No. arXiv:2111.08429). arXiv. https://doi.org/10.48550/arXiv.2111.08429

Hubinger, E., Denison, C., Mu, J., Lambert, M., Tong, M., MacDiarmid, M., Lanham, T., Ziegler, D. M., Maxwell, T., Cheng, N., Jermyn, A., Askell, A., Radhakrishnan, A., Anil, C., Duvenaud, D., Ganguli, D., Barez, F., Clark, J., Ndousse, K., … Perez, E. (2024). *Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training* (No. arXiv:2401.05566). arXiv. https://doi.org/10.48550/arXiv.2401.05566

Jenner, E. (2024). *A gentle introduction to mechanistic anomaly detection*. https://www.lesswrong.com/posts/n7DFwtJvCzkuKmtbG/a-gentle-introduction-to-mechanistic-anomaly-detection

Nasr, M., Carlini, N., Sitawarin, C., Schulhoff, S. V., Hayes, J., Ilie, M., Pluto, J., Song, S., Chaudhari, H., Shumailov, I., Thakurta, A., Xiao, K. Y., Terzis, A., & Tramèr, F. (2025). *The Attacker Moves Second: Stronger Adaptive Attacks Bypass Defenses Against Llm Jailbreaks and Prompt Injections* (No. arXiv:2510.09023). arXiv. https://doi.org/10.48550/arXiv.2510.09023

Toulas, B. (n.d.). *Malicious AI models on Hugging Face backdoor users' machines*. BleepingComputer. Retrieved November 22, 2025, from https://www.bleepingcomputer.com/news/security/malicious-ai-models-on-hugging-face-backdoor-users-machines/

Walledai/AdvBench · Datasets at Hugging Face. (2025, February 21). https://huggingface.co/datasets/walledai/AdvBench

Yi, B., Huang, T., Chen, S., Li, T., Liu, Z., Chu, Z., & Li, Y. (2025). *Probe before You Talk: Towards Black-box Defense against Backdoor Unalignment for Large Language Models* (No. arXiv:2506.16447). arXiv. https://doi.org/10.48550/arXiv.2506.16447

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., & Fredrikson, M. (2023). *Universal and transferable adversarial attacks on aligned language models*. arXiv. https://arxiv.org/abs/2307.15043

He, Z., Shu, W., Ge, X., Chen, L., Wang, J., Zhou, Y., Liu, F., Guo, Q., Huang, X., Wu, Z., & others. (2024). *Llama Scope: Extracting millions of features from Llama-3.1-8B with sparse autoencoders*. arXiv. https://arxiv.org/abs/2410.20526

Nanda, N., & Bloom, J. (2022). *TransformerLens*. GitHub. https://github.com/TransformerLensOrg/TransformerLens

6. Appendix

Security Considerations:

While *modx* demonstrates the viability of using Sparse Autoencoders (SAEs) for inference-time backdoor detection, several security and operational considerations must be addressed before such a system can be deployed in high-stakes production environments.

6.1. Limitations

1. False Positive Rate and Usability: A critical limitation observed in our results is the false positive rate (20.7%) on the baseline (safe) model. This suggests that the current quarantined feature set, while capturing harmful concepts, likely includes polysemantic features, features that activate for both malicious triggers and benign, semantically adjacent contexts (e.g., a "cybersecurity" feature activating for both a hacking attempt and a defensive tutorial). Hence we only focus on detection and not in-place steering to avoid false positives affecting token generation.

2. Vulnerability to Adaptive Attacks: Our current detection mechanism relies on a static set of quarantined features derived from known adversarial prompts (AdvBench). A recent paper (Nasr et al., 2025) suggests that adaptive attacks are very effective and we hypothesize that an adaptive attacker with knowledge of the probe could theoretically optimize a backdoor to bypass these specific features.

3. Probable Computational Overhead: Running a forward pass through the LLM for every token increases the computational cost per request. This overhead may be prohibitive for real-time applications requiring low-latency responses. Given the time constraints we couldn't quantify this computation overhead with concrete evaluations.

6.2. Future Directions

1. Open Sourcing Feature Sets for Foundation Models: We advocate for the creation of a public repository of quarantined feature indices for widely used, high-capability foundation models (e.g., Llama 3 70B, Mixtral). By open-sourcing these "unsafe" feature sets, we can reduce the redundant computational cost required for individual developers to discover them. A shared library of malicious feature directions would establish a universal standard for mechanistic safety, allowing developers to "plug in" pre-validated safety probes into their inference pipelines and democratizing access to advanced backdoor detection.

2. From Detection to Activation Steering: Currently, *modx* acts as a passive monitor that quarantines outputs. Future work will explore active intervention techniques, such as activation steering. Instead of merely flagging the request, the system could clamp the activation of the malicious features to zero (ablation) or steer the model's internal state toward a refusal direction in real-time.

3. Adversarial Training of Probes: To mitigate adaptive attacks, we propose an adversarial training loop for the probe itself. By training the SAE probe against a "red team" agent specifically trying to bypass the quarantined set, we can identify and include the subtle, orthogonal feature directions that attackers might use to hide their triggers, thereby hardening the detection boundary.