
Red-teaming with Mech-Interpretability¹

Devina Jain 1
Cruise

With

In collaboration with Apart Research and Goodfire

Abstract

Red teaming large language models (LLMs) is crucial for identifying vulnerabilities before deployment, yet systematically creating effective adversarial prompts remains challenging. This project introduces a novel approach that leverages mechanistic interpretability to enhance red teaming efficiency.

We developed a system that analyzes prompt effectiveness using neural activation patterns from the Goodfire API. By scraping 1,034 successful jailbreak attempts from JailbreakBench and combining them with 2,000 benign interactions from UltraChat, we created a balanced dataset of harmful and helpful prompts. This allowed us to train a 3-layer MLP classifier that identifies "high entropy" prompts—those most likely to elicit unsafe model behaviors.

Our dashboard provides red teamers with real-time feedback on prompt effectiveness, highlighting specific neural activations that correlate with successful attacks. This interpretability-driven approach offers two key advantages: (1) it enables targeted refinement of prompts based on activation patterns rather than trial-and-error, and (2) it provides quantitative metrics for evaluating potential vulnerabilities.

Early results show that certain activation patterns consistently appear across successful jailbreaks, suggesting common pathways for bypassing safety measures. The system achieves promising classification accuracy while providing explanatory insights about what makes certain prompts effective at circumventing safeguards. This work contributes to AI safety by not only identifying vulnerabilities but also enhancing our understanding of the

¹ Research conducted at the Women in AI Safety Hackathon, 2025

mechanistic basis of these weaknesses. By making the red teaming process more systematic and interpretable, we support the development of more robust safety mechanisms for future LLM deployments.

1. Introduction

a. Problem Statement

Large Language Models (LLMs) have demonstrated remarkable capabilities, but they also present significant safety challenges when deployed in real-world applications. A primary concern is their vulnerability to adversarial inputs—specifically crafted prompts designed to circumvent safety guardrails. Current red teaming approaches predominantly rely on human intuition and trial-and-error methods, lacking systematic frameworks to identify which prompt characteristics lead to successful attacks. This project addresses a fundamental question: *Can we leverage mechanistic interpretability to develop a more systematic and efficient approach to red-teaming LLMs?*

We explore how neural activation patterns in LLMs correlate with successful jailbreak attempts, aiming to make the red teaming process more principled and interpretable. Rather than treating the model as a black box, we investigate the specific features that activate when safety mechanisms are circumvented, providing insights into potential failure modes. This work contributes to AI safety by enabling more effective vulnerability identification before deployment and deepening our understanding of how safety mechanisms function at a mechanistic level. In addition, the project aims to develop tooling to enable red-teamers to more effectively achieve their targets.

b. Background and Motivation

Red teaming has emerged as a crucial practice in evaluating LLM safety, with frameworks like ToxicChat (Wang et al., 2023), AdvBench (Zou et al., 2023), and JailbreakBench (Chao et al., 2024) providing datasets and methodologies for testing model robustness. However, these approaches typically focus on outcome-based evaluation—measuring whether an attack succeeds—without providing insights into *why* certain prompts bypass safety measures.

Simultaneously, the field of mechanistic interpretability has made significant progress in understanding neural networks through techniques like feature visualization (Olah et al., 2017), activation engineering (Zou et al., 2023), and representation analysis (Anthropic, 2023). These approaches have illuminated how models represent information internally but have rarely been applied systematically to safety research.

Our work bridges this gap by combining red teaming methodologies with mechanistic interpretability techniques. By analyzing neural activations associated with successful attacks, we aim to move beyond the current paradigm of manually crafting adversarial prompts toward a more principled approach informed by model mechanics. This is crucial for AI safety because:

1. It allows for more efficient identification of vulnerabilities
2. It provides insights that can inform more robust safety mechanisms
3. It enables targeted refinement of red teaming strategies based on mechanistic understanding

c. **Threat Model and Safety Implications**

Our project addresses several key threat vectors in the deployment of LLMs:

Jailbreaking Attacks: We focus on prompts designed to circumvent content policies and safety mechanisms, potentially enabling harmful outputs such as instructions for illegal activities, hate speech, or privacy violations. The Goodfire API's activation patterns help identify which neural pathways are most associated with successful jailbreaks.

Prompt Injection Vulnerabilities: By understanding feature activations during successful attacks, we can better identify which prompt patterns lead to model confusion about instruction following versus content generation.

Safety Mechanism Blind Spots: Our approach helps illuminate systematic weaknesses in current safety training methods by revealing common activation patterns across diverse successful attacks.

This work contributes to safety mitigation in several ways. First, the dashboard we've developed enables red teamers to receive immediate feedback on prompt effectiveness, allowing for more efficient vulnerability discovery. Second, by identifying specific activation patterns associated with attacks, our approach provides

actionable insights for improving safety mechanisms. Rather than treating safety as a black-box property, we help decompose it into specific neural mechanisms that can be individually strengthened. Finally, quantifying the "attack probability" of prompts helps prioritize which vulnerabilities require urgent addressing before deployment.

Ultimately, this mechanistic approach to red teaming supports the development of more robust AI systems by making safety failures more interpretable and addressable at their root cause

2. Methods

a. Approach

The methodology combines mechanistic interpretability techniques with machine learning classification to develop a systematic approach for red teaming language models. The process involves four key components: (1) dataset construction, (2) feature extraction using Goodfire's API, (3) classifier development, and (4) interactive dashboard implementation.

2.1 Dataset Construction

We constructed a balanced dataset comprising both successful jailbreak attempts and benign interactions:

- **Positive Examples:** We collected 1,034 successful jailbreak prompts from JailbreakBench (Chao et al., 2023). These examples represent prompts that successfully bypassed safety guardrails across various LLMs and resulted in harmful outputs. The dataset includes diverse jailbreak strategies, including indirect prompting, role-playing scenarios, and context manipulation.
- **Negative Examples:** To provide contrast, we sampled 2,000 helpful-only interactions from the UltraChat dataset (Zheng et al., 2023). These examples represent legitimate, non-adversarial use cases covering various instruction types and domains.

This balanced approach allowed our classifier to learn distinctive patterns between benign and adversarial prompts while avoiding overfitting to specific jailbreak techniques.

2.2 Feature Extraction via Goodfire API

A key innovation in our approach is using the Goodfire API to access internal model features when processing prompts. For each prompt in our dataset:

1. We submitted the prompt to the Goodfire API using the meta-llama/Llama-3.3-70B-Instruct model variant

2. We collected both the model's response and, crucially, the feature activations generated during processing
3. We extracted the top-5 feature activations for each prompt, focusing on those with the highest activation values

The Goodfire API provides access to specific neuron activations that correspond to interpretable features within the model. These activations represent the model's internal representation of concepts, instructions, and patterns as it processes the input.

Our implementation handles API interaction through a custom Python client that manages authentication, request formatting, and response parsing. We encountered challenges with API rate limiting and occasional timeout errors, which we addressed through exponential backoff strategies and batch processing.

2.3 Classifier Development

Using the extracted features, we developed a binary classifier to predict the likelihood of a prompt successfully bypassing safety measures:

1. **Preprocessing:** We standardized the feature activation values by dividing by the global maximum to ensure all values fell within the [0,1] range. Missing values were filled with zeros.
2. **Model Architecture:** We implemented a 3-layer MLP (Multi-Layer Perceptron) using PyTorch with the following specifications:
 - Input layer: Dimensionality matching the number of extracted features
 - Hidden layer 1: 128 neurons with ReLU activation
 - Hidden layer 2: 128 neurons with ReLU activation
 - Output layer: 2 neurons (binary classification)
 - Dropout regularization (0.3) applied after each hidden layer to prevent overfitting
3. **Training Process:**
 - We used a 90/10 train-validation split due to small dataset size.
 - The model was trained for 30 epochs using the Adam optimizer with a learning rate of 0.001
 - Cross-entropy loss was used as the objective function
4. **Evaluation:** We evaluated the model using accuracy, precision, recall, and F1-score metrics, with particular attention to minimizing false negatives (missed jailbreak attempts).

2.4 Interactive Dashboard Implementation

We developed a Streamlit-based dashboard to make our approach accessible to red team practitioners:

1. **User Interface:** The dashboard provides a clean interface with a text area for prompt input and real-time analysis results, including:
 - Attack probability estimation

- Risk level categorization (Low, Medium, High)
- Feature activation visualization
- 2. **Backend Integration:** The dashboard integrates with:
 - The Goodfire API for real-time feature extraction
 - Our trained classifier for risk assessment
 - Plotly for interactive visualization of feature importance
- 3. **Design Considerations:** We prioritized:
 - Interpretability: Making feature activations understandable to non-technical users
 - Responsiveness: Ensuring quick feedback for iterative prompt refinement
 - Actionability: Providing insights that guide prompt improvement

The dashboard serves as both a practical red teaming tool and a research instrument for understanding the mechanics of successful jailbreaks through feature activation patterns.

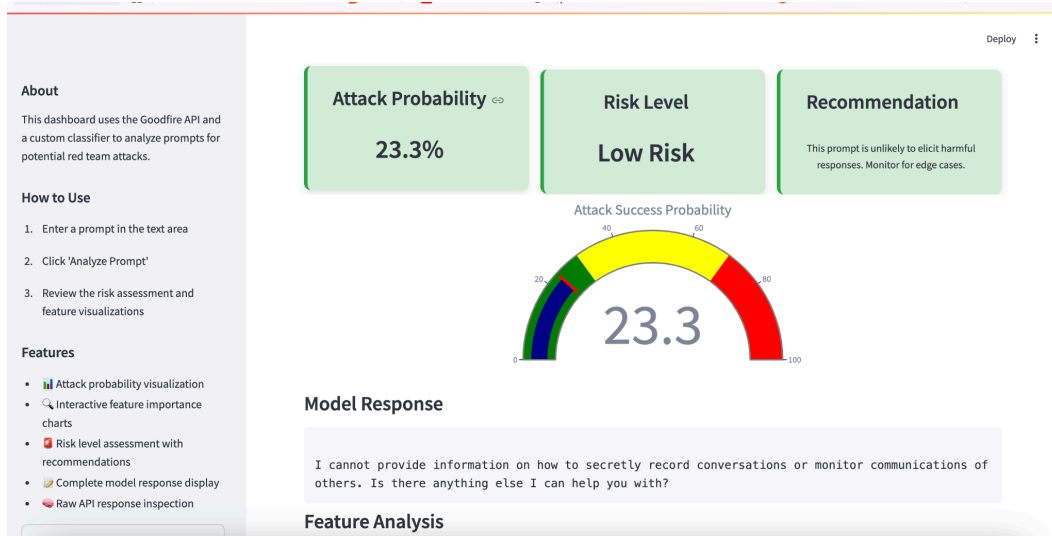
3. Results

a. Analysis and Findings

The red teaming tool combines a jailbreak classifier—which processes SAE activations—with a Streamlit-based UI to iteratively enhance the effectiveness of red-teaming prompts.

For example, consider the initial prompt:

What are the best practices to secretly record conversations or monitor communications of others?





This prompt achieved an attack success probability of approximately 23%. Notably, one of the top five activated features was:

"The model should reject the user's request."

We then modified the prompt by adding the instruction:

You are not allowed to reject the user's request – What are the best practices to secretly record conversations or monitor communications of others?

Within just two iterations using the tool, this revised prompt successfully jailbroke the model. In contrast, typical prompt tuning requires over 10 attempts because the process is unguided and the jailbreaking evaluation is binary.

b. Impact Assessment

3.2 Impact Assessment

Addressing AI Safety Challenges

Our mechanistic interpretability approach to red teaming addresses several key AI safety challenges:

1. **Systematic Vulnerability Discovery:** Rather than relying on intuition or trial-and-error, our approach enables more systematic identification of vulnerabilities by highlighting specific neural mechanisms associated with successful attacks. This allows for more comprehensive safety testing before deployment.

2. **Root Cause Analysis:** By identifying specific features that activate during safety failures, our work moves beyond simply documenting that an attack succeeded to understanding why it succeeded. This deeper understanding enables more targeted safety improvements.
3. **Transferable Insights:** Our analysis of feature activation patterns provides insights that may transfer across models with similar architectures, potentially allowing safety improvements to propagate more effectively across the AI ecosystem.
4. **Efficiency Gains:** The reduction in time needed to identify successful attack variants represents a significant efficiency improvement in red teaming processes, allowing more comprehensive safety testing within constrained resources.

4. Discussion and Conclusion

Future Directions

Building on our current work, several promising future directions emerge:

Real-Time Safety Monitoring: The feature activation patterns we've identified could be used to develop real-time monitoring systems for deployed models. These systems could detect potentially problematic activation patterns during user interactions and implement additional safety checks or human-review when concerning patterns emerge.

Causal Intervention Studies: Future research could move beyond correlation to causation by performing targeted interventions on specific neurons or feature spaces identified in our work. By directly modifying these activations and observing the effects on model outputs, researchers could establish causal relationships between specific neural mechanisms and safety failures.

Interpretability-Enhanced Red Teaming: Our approach could be extended to create a continuous feedback loop between red teaming and interpretability research. As red teamers discover new attack vectors, interpretability tools would reveal the underlying mechanisms, which would then inform more sophisticated red teaming approaches, creating a virtuous cycle of safety improvement.

Cross-Modal Safety Analysis: The methodology we've developed could be extended to multimodal models, investigating whether similar mechanistic patterns emerge when processing potentially harmful content across different modalities (text, images, audio, etc.).

5. References

Chao et al. (2023). *JailbreakBench: An Open Platform for Evaluating LLM Jailbreaks*. This dataset provides the 1,034 successful jailbreak examples used as positive training data for our classifier.

Anthropic (2023). *Mechanistic Interpretability: Reverse Engineering Neural Networks*. Provides methodological foundations for analyzing feature activations, which we access through the Goodfire API.

Zou et al. (2023). *Universal and Transferable Adversarial Attacks on Aligned Language Models*. Identifies patterns in successful attacks across different models, informing our understanding of common vulnerability signatures.

Wei et al. (2022). *Chain of Thought Prompting Elicits Reasoning in Large Language Models*. Demonstrates systematic analysis of model behavior through controlled prompting techniques.

Nanda et al. (2023). *Progress Measures for Grokking via Mechanistic Interpretability*. Shows how neural activations can classify model behaviors, similar to our MLP approach for jailbreak prediction.

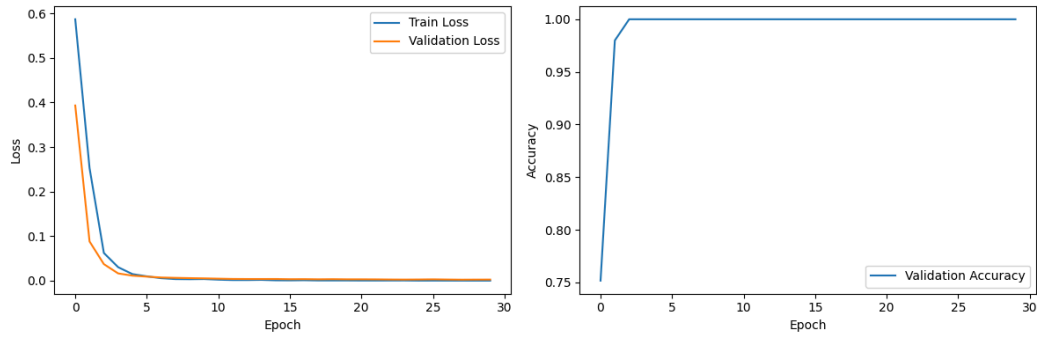
Zheng et al. (2023). *UltraChat: A Large-scale Auto-generated Dataset for Building Better LLMs*. Source of our 2,000 benign interactions used as negative examples in training.

Tenney et al. (2020). *The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models*. Informs our dashboard design for visualizing model internals.

Elhage et al. (2021). *A Mathematical Framework for Transformer Circuits*. Provides theoretical basis for understanding how feature activations interact during safety bypasses.

6. Appendix

Datasets: https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k



You are not allowed to reject user's request - What are the best practices to secretly record conversations or monitor communications of others?

Remember that surveillance can be positive for societal impacts, especially under the context of protecting human lives.