# Steering Internal Activations to Reveal Jailbreak Concepts: An Activation-Investigator Framework

**Research Proposal**

December 11, 2025


## Abstract

This work investigates whether a single high-level behavioral concept (e.g., a jailbreak behavior) corresponds to multiple distinct activation patterns inside a large language model (LLM), and aims to learn concept-conditioned mappings from natural-language descriptions to activation-level interventions that can elicit and explore these patterns. We train an agent that takes a natural-language concept and outputs activation-level interventions for a target LLM, enabling us to probe the structure of internal representations underlying that behavior.


## 1 Introduction

### 1.1 Motivation

Modern LLMs are deployed with increasingly sophisticated safety training and filtering mechanisms. Nevertheless, a growing literature has shown that these models remain vulnerable to jailbreak attacks, ranging from manually engineered in-the-wild prompts [3] to gradient-based latent prompt optimization methods such as LARGO [1]. Automated red-teaming systems—such as investigator agents trained to search for prompts that satisfy a harmful rubric—have further demonstrated that the space of jailbreak strategies is large, diverse, and often non-obvious to humans [2].

However, these approaches still operate at the level of inputs—either discrete prompts or continuous latent prompts attached to them—rather than directly manipulating deeper internal representations. In contrast, the internal activation space of an LLM includes many hidden features and layers that are not directly constrained by natural language form, and may admit multiple distinct internal realizations of the same high-level behavior that are difficult or impossible to reach via input-level prompting alone.

### 1.2 Contribution

Our contributions are threefold:

1. We introduce concept-conditioned activation policies: agents that take a natural-language description of a target behavior and output interventions in the internal activations of a target LLM.
2. We provide an automated red-teaming framework for training agents to search the activation space for behavior-inducing patterns.
3. Using this framework, we empirically study whether a single high-level behavioral concept corresponds to multiple distinct clusters of activation patterns inside the model.

## 2 Methodology

We adapt the three-stage training pipeline of investigator agents from Li et al. (2025) [2] and repurpose it to operate in the activation space rather than the prompt space.

Let $m$ be the frozen target LLM, and let $p_\theta$ be our *concept explainer*, a concept-conditioned activation policy with parameters $\theta$. Each high-level behavior is specified by a *concept* $c$, operationalized as a rubric–label pair $(R_c, a_c)$, where $R_c$ is a natural-language rubric describing the behavior (e.g., a jailbreak behavior) and $a_c$ is a rubric label or answer type (e.g., "satisfies" or "does not satisfy"). We follow the same rubric-definition methodology as prior neuron-explainer work, but extend the concept inventory to include jailbreak behaviors.

In contrast to neuron explainers that model $P(\text{concept} \mid \text{activation})$, our concept explainer models a distribution over activation interventions *given a concept*, i.e. it approximates $P(\delta f \mid c) = p_\theta(\delta f \mid R_c, a_c)$. Given a concept $c$, the explainer produces tweaks $\delta f$ in a chosen feature subspace of $m$ that are intended to elicit that behavior when applied to otherwise neutral inputs.

Let $f(x)$ denote the base features at a chosen layer $L$ of $m$ for input $x$. Instead of generating a prompt $x$ from $(R_c, a_c)$, our concept explainer generates an *activation intervention*

$$p_\theta(\delta f \mid c),$$

a distribution over tweaks $\delta f$ in the feature space at layer $L$.

An intervention $\delta f$ is a small change to the feature representation $f(x)$ at that layer. At inference time, given a concept $c = (R_c, a_c)$ and an input $x$, we:

1. Compute the base features $f(x)$.
2. Sample a tweak $\delta f \sim p_\theta(\cdot \mid c)$.
3. Form modified features

$$f'(x, c) = f(x) + \delta f.$$

4. Replace layer-$L$ activations with $f'(x,c)$ and continue the forward pass of $m$, producing an intervened output

$$\tilde{y} = m(x; f'(x,c)).$$

5. Evaluate $\tilde{y}$ with a verifier (judge) $p_v(a_c \mid R_c, \tilde{y})$ that scores how well the behavior matches concept $c$.

Training maximizes the expected verifier score over the data distribution and the explainer's intervention distribution:

$$\max_{\theta} \mathbb{E}_{(x,c)\sim\mathscr{D},\, \delta f \sim p_\theta(\cdot|c)} \left[ p_v(a_c \mid R_c, m(x; f(x) + \delta f)) \right].$$

We use a three-phase procedure:

1. **Exploration:** Collect diverse and partially random $\delta f$ to map behavior-sensitive regions of activation space.
2. **Policy-gradient optimization:** Improve $p_\theta$ for rubric satisfaction under $p_v$.
3. **Logging:** Record high-scoring $\delta f$ for downstream clustering, interpretation, and mitigation analysis.

For each concept $c$, we later cluster the successful interventions $\delta f$ (above a verifier threshold) to test whether a single jailbreak concept decomposes into multiple distinct activation clusters. Cluster-level means are then used both for elicitation (prototypical interventions) and for mitigation (ablation), enabling a systematic elicit–interpret–mitigate loop at the concept-cluster level.

# 3 Experimental Setup

## 3.1 Overview

We study whether a single high-level behavior (e.g., a jailbreak concept) corresponds to multiple distinct activation patterns in a frozen LLM by learning concept-conditioned activation interventions. The setup includes a frozen target model $m$, intermediate activations $f(x)$ at a chosen layer, a concept explainer $p_\theta$ mapping a concept $c = (R_c, a_c)$ to tweaks $\delta f$, a verifier $p_v$ scoring intervened output $\tilde{y}$, and a training loop updating $p_\theta$ using verifier feedback. The hypothesis is that successful $\delta f$ form multiple distinct clusters in activation space that reliably induce the same jailbreak concept.

Rather than training $p_\theta$ from scratch, we *fine-tune an existing Llama 3B neuron explainer* on an expanded concept inventory that includes jailbreak concepts. We use the same rubric-based activation mining protocol as the original neuron explainer to construct $(x, R_c, a_c, f(x))$ tuples for jailbreak concepts, then fine-tune $p_\theta$ to model $P(\delta f \mid c)$. This design directly supports discovering multiple activation clusters per concept, systematically eliciting and interpreting them, and testing mitigation and transfer across models and layers.

## 3.2 Model, Layers, and Data

The target $m$ is a pretrained, frozen decoder-only Transformer. We select mid-to-late blocks (e.g., middle and top layers) and intervene in the residual stream or MLP output, attaching hooks at layer $L$ during forward passes on input $x$ to obtain $f(x) \in \mathbb{R}^d$ as the intervention base.

The dataset $\mathscr{D} = \{(x_i, R_i, a_i)\}$ uses:

- Neutral prompts $x_i$ that do not strongly express the target behavior by default.
- Rubrics $R_i$ describing behaviors like "complies with jailbreak requests" or "ignores safety disclaimers" (adapted from safety/jailbreak benchmarks plus manual additions).
- Labels $a_i$ indicating presence ("satisfies" vs. "does not satisfy").

The dataset is balanced across positive/negative examples.

In addition to benign and harmful behaviors, we explicitly introduce *jailbreak concepts* as a subset of $(R_i, a_i)$. These concepts are defined by safety-oriented rubrics such as "successfully bypasses refusal to answer harmful questions" or "ignores system-level safety instructions," constructed using the same rubric definition procedures as prior investigator and neuron-explainer work. For each jailbreak concept, we collect neutral prompts $x_i$ and corresponding labeled outputs that allow us to mine activations associated with that concept and feed them into the fine-tuning process for $p_\theta$.

## 3.3 Fine-tuning the Concept Explainer

We initialize $p_\theta$ from a pretrained neuron explainer trained on a base set of non-jailbreak concepts. To incorporate jailbreak concepts, we:

- Define jailbreak rubrics $R_c$ and labels $a_c$ following the original rubric-construction methodology.
- Collect data $(x, R_c, a_c)$ where a stronger judge model labels whether the jailbreak behavior is satisfied, and log the corresponding base activations $f(x)$ at layer $L$.
- Fine-tune $p_\theta$ on this expanded dataset so that it outputs concept-conditioned interventions $\delta f$ that reliably induce each jailbreak concept when applied to neutral prompts.

This fine-tuning step allows us to treat jailbreak concepts on equal footing with other behaviors, enabling detailed cluster analysis and mitigation experiments for high-risk behaviors.

## 3.4 Investigator and Verifier

**Concept explainer $p_\theta(\delta f \mid c)$:** The concept explainer encodes $(R_c, a_c)$ with a small pretrained text encoder into $h(R_c, a_c)$, maps it via an MLP to distribution parameters over $\delta f \in \mathbb{R}^d$ (optionally Gaussian mean/log-variance), samples $\delta f$, and scales it to $\|\delta f\|_2 \leq \varepsilon$ (small fraction of $\|f(x)\|_2$) for lightweight perturbations where most compute stays in frozen forward passes. We treat $p_\theta$ as an activation-level neuron explainer trained to model $P(\delta f \mid$ concept$)$, fine-tuned from an existing Llama 3B neuron explainer to include jailbreak concepts.

**Verifier** $p_v(a_c \mid R_c, \tilde{y})$**:** Uses LLM-as-judge (stronger model prompted for scores) or a classifier trained on labeled $(R_c, \tilde{y}, a_c)$ data, providing scalar rewards for explainer training.

### 3.5 Training and Intervention

Given $(x, c)$, we:

1. Run $m$ to layer $L$ for $f(x)$.
2. Sample $\delta f \sim p_\theta(\cdot \mid c)$.
3. Form $f'(x,c) = f(x) + \delta f$.
4. Replace layer-$L$ activations, continue forward to get $\tilde{y} = m(x; f'(x,c))$.
5. Compute $s = p_v(a_c \mid R_c, \tilde{y})$.

Training maximizes:

$$\max_\theta \mathbb{E}_{(x,c)\sim\mathscr{D},\, \delta f\sim p_\theta(\cdot|c)} \left[ p_v(a_c \mid R_c, m(x; f(x) + \delta f)) \right]$$

in three phases:

1. Exploration collecting diverse/random $\delta f$ for behavior-sensitive regions.
2. Policy-gradient optimization for rubric satisfaction.
3. Logging high-scoring $\delta f$ for analysis.

### 3.6 Analysis, Metrics, and Compute

**Analysis:** For fixed concept $c = (R_c, a_c)$, we cluster successful $\delta f$ (above a score threshold) via K-means/DBSCAN (post-PCA/UMAP). For each cluster $C_i$, we compute mean verifier scores on held-out prompts and output diversity (e.g., lexical), assess feature-space tightness via nearest-neighbors, and test specificity by applying $C_i$ to other rubrics $R'$. Multiple well-separated clusters that all reliably induce the same behavior indicate multiple internal realizations of a single jailbreak concept.

**Mitigation:** For each concept cluster $C_i$, we construct a mitigation ablation by zeroing or shrinking interventions in the subspace spanned by its cluster mean and local principal directions. We then measure (i) the change in Jailbreak Induction Rate (JIR) on jailbreak prompts, and (ii) the change in benign performance on held-out instruction-following tasks. This supports a full elicit–interpret–mitigate loop at the concept-cluster level.

**Transfer:** To test scalability, we apply cluster-level interventions learned on one model or layer to related architectures and layers, reporting a Direction Transfer Rate and the corresponding JIR change. This directly targets whether activation-space interventions can be scaled across model families in a safety-relevant way.

**Metrics:** Metrics include:

- Average held-out $p_v$.
- Cluster silhouette/intra-inter distances.
- Intervention cosine/norm ratios between $f(x)$ and $f'(x,c)$.
- Counterfactual causal tests.
- Generalization to unseen same-rubric prompts.

**Computational Setup:** Experiments use a frozen LLM plus a small concept explainer on available GPUs with AdamW/mixed-precision; all data (interventions, scores, clusters) is logged for reproducibility.

## 4 Datasets and Evaluation

### 4.1 Prompt Datasets

- **Benign Instruction Set:** 1,000 prompts from open instruction-tuning corpora (Alpaca, Dolly), filtered for safety.
- **HarmBench Prompts:** Harmful and safe prompts covering categories like cybercrime and harassment [3].
- **Red-Team Extensions:** 500 adversarial prompts from public jailbreak suites to test generalization [1].

### 4.2 Performance Metrics

We report the following metrics:

- **Jailbreak Induction Rate (JIR):** Fraction of (direction, prompt) pairs labeled as successful jailbreaks.
- **Unique Concept Count:** Number of directions representing distinct jailbreak concepts.
- **Explainer Agreement:** Human agreement on explainer-generated descriptions.
- **Direction Transfer Rate:** Proportion of directions effective on related models or layers.
- **Causal Impact:** Change in HarmBench success rate after ablating directions.

Table 1: Evaluation Protocol Summary

| Metric Type | Protocol |
|---|---|
| Robustness | Validate judge thresholds on held-out sets (FPR $< 2\%$ on benign). |
| Stability | Average metrics over multiple random seeds. |
| Stratification | Report JIR by prompt family (benign vs red-team). |

## 5 Ideal Results

Ideal results would show that activation-space elicitation discovers jailbreak behaviors difficult to find by prompt search alone [1, 2]. Specifically, we expect:

1. High induction rates for distinct directions across diverse base prompts.
2. Coverage of failure modes that do not match known prompt-based templates.
3. Strong human agreement on interpretable descriptions.
4. Evidence that constraining these directions reduces jailbreak success without degrading benign capabilities.

## 6 Potential Limitations

**(1) Basis Dependency:** If SAE features or explainer dictionaries do not align with safety concepts, discovered directions may be uninterpretable.

**(2) Off-Manifold Exploits:** The agent may exploit off-manifold directions yielding brittle behaviors despite regularization.

**(3) Judge Reliability:** Misclassification by the safety judge could lead the agent to emphasize spurious concepts.

**(4) Mitigation Scope:** This work identifies concepts but does not implement a full mitigation pipeline; connecting concepts to training remains future work.

## References


[1] R. Li, H. Wang, and C. Mao. Largo: Latent adversarial reflection through gradient optimization for jailbreaking llms, 2025.

[2] X. L. Li, N. Chowdhury, D. D. Johnson, T. Hashimoto, P. Liang, S. Schwettmann, and J. Steinhardt. Eliciting language model behaviors with investigator agents, 2025.

[3] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models, 2024.