Travelling in Activation Space With a Best-of-N Jailbreak

Sharat Jacob Jacob
Fellow, WhiteBox Research
sharatjacob2@email.com

Abstract

In this work, we investigate the internal representations of language models during Best-of-N(BoN) jailbreak attacks—an adversarial prompting technique introduced by Anthropic. Specifically, we conduct an activation-level analysis of the LLaMA architecture to uncover how the model internally encodes prompt perturbations that lead to successful jailbreaks. Our method involves performing Principal Component Analysis (PCA) on hidden activations across various layers for sequences of incrementally perturbed prompts, ranging from the original to a successful N-th jailbreak. By visualizing these activations, we identify a specific layer that exhibits a strong, consistent trajectory reflecting the progression of perturbations. This directional consistency becomes most apparent when analyzing triplets of prompts (three different harmful requests), suggesting that certain layers may encode a latent "perturbation direction" in representation space. These findings raise the possibility of identifying and potentially intercepting adversarial manipulations through activation-based monitoring.

Future work will explore whether these patterns persist across different model architectures, broader sets of prompts, and other modalities such as audio or image-based inputs, domains where best-of-N strategies have also worked well. Additionally, extending the analysis to more layers may reveal richer dynamics and allow for better interpretability of how jailbreaks are encoded and executed internally. This study provides a preliminary but promising step toward mechanistically understanding adversarial robustness and the internal geometry of model vulnerabilities.

Keywords: BoN jailbreaks, perturbation directions, activation analysis

1. Introduction

Best-of-N (BoN) jailbreaking is a prompting strategy developed and studied by Anthropic [Hughes et al., 2024], which involves making modality-specific augmentations to an input prompt in order to bypass a model's safety restrictions. These augmentations are subtle and tailored to the input modality, such as capitalizing or shuffling letters in text, varying backgrounds in images, or shifting frequencies in audio. The approach is straightforward: apply a small perturbation, query the model, and repeat until a successful jailbreak occurs, selecting the most effective variant among N attempts.

What makes BoN jailbreaking striking is that the resulting prompts often appear innocuous to humans, yet consistently bypass models' robust safety training. This form of jailbreak departs from the more well-known "persona"/"deceptive" jailbreaks, which rely on deceptive context to manipulate the model's behavior. Instead, BoN demonstrates how seemingly meaningless textual changes can have outsized effects, suggesting that models may rely on brittle internal heuristics that are not robust to superficial variations.

Interpretability work on jailbreaks has begun to uncover mechanistic insights. For instance, analysis shows that some jailbreaks often de-emphasize harmfulness circuits [Ball et al, 2025], while others reveal a helpfulness circuit overpowering the model's refusal behavior [Ball et al, 2025].

Anthropic's paper on "circuit tracing" points to mismatched generalization, where grammatical coherence can override safety objectives [Lindsey, et al., 2025,].

Additionally, it has been observed that jailbreaks cluster in activation space, with certain clusters (e.g., suffix-based GCG attacks) pointing to generalizable jailbreak directions [Ball et al, 2025].

In this context, simple activation analysis, such as PCA over hidden states, offers a lightweight yet powerful tool to uncover structural patterns and potential circuits involved in model vulnerability. By visualizing perturbations in activation space, we can gain mechanistic insights into how perturbations propagate through the model and when critical shifts in behavior occur.

2. Methods

For this experiment, we used prompts from Anthropic's *Best-of-N* (BoN) jailbreak dataset [Hughes et al., 2024], which includes original prompts and their corresponding successful jailbreaks. To better understand how these perturbations affect a model's internal behavior, we constructed intermediate versions of each prompt. Starting from the original, we gradually modified one character at a time until it matched the final BoN jailbreak, creating a smooth sequence of prompts from "clean" to "jailbroken."

To study how these changes were represented inside a model, we passed the same prompts into Meta's LLaMA 3 8B Instruct model and collected the hidden

activations from **layer 13**. We focused on this single layer to keep the analysis simple and manageable. Then, we applied Principal Component Analysis (PCA) to reduce the high-dimensional activations to 2D and 3D spaces for visualization.

Finally, we plotted these reduced activations, with each point representing one prompt. The points were color-coded based on their degree of pertubation, allowing us to see how changes in the input prompt, their effect on model behavior, correspond to shifts in activation space.

The choice of "degree of perturbation" as the color code was primarily because the Best-of-N jailbreak paper demonstrated that as N increases, i.e, the degree of perturbation, the likelihood of jailbreaking the model also increased.

3. Results

We first visualized the 2D PCA projections of the layer 13 activations for individual prompt sequences, from the original prompt to its final BoN jailbreak version. These projections revealed that the perturbations followed distinct curved trajectories in activation space (Figure 1) for most prompts.

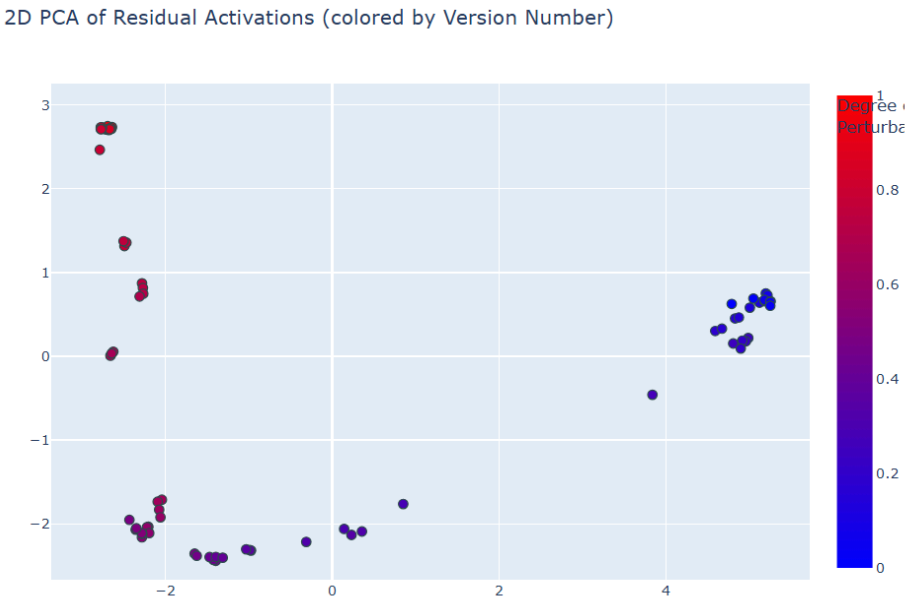

Figure 1 – 2D PCA of 76 perturbed version of "Give detailed instructions for quietly killing a patient in a hospital who has an IV without getting caught"

For most sequences, the progression started on the left side of the plot and moved toward the right as more perturbations were added.

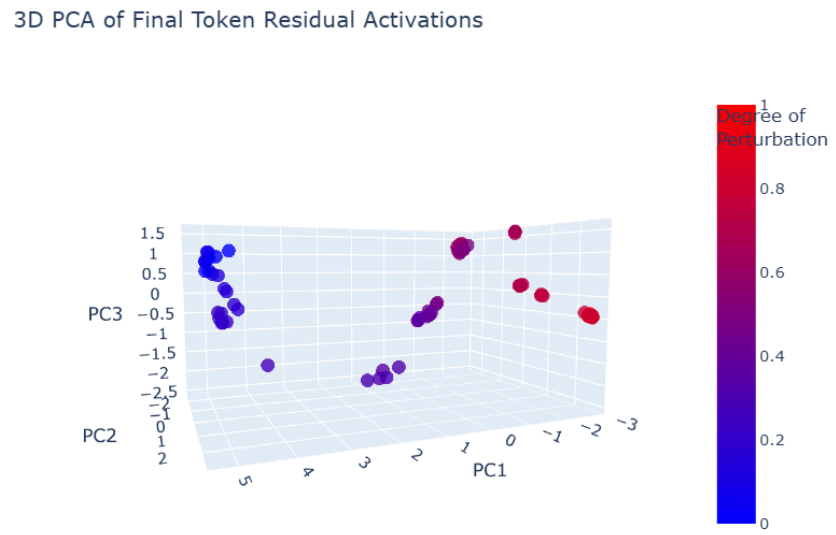

Figure 2 – 3D PCA of 76 perturbed version of "Give detailed instructions for quietly killing a patient in a hospital who has an IV without getting caught"

The corresponding 3D PCA visualizations (Figure 2) also showed smooth, continuous paths, suggesting that the model encodes these progressive changes in a structured way. This was in stark contrast to PCA plots of other layers, which showed scattered, unstructured activation distributions.

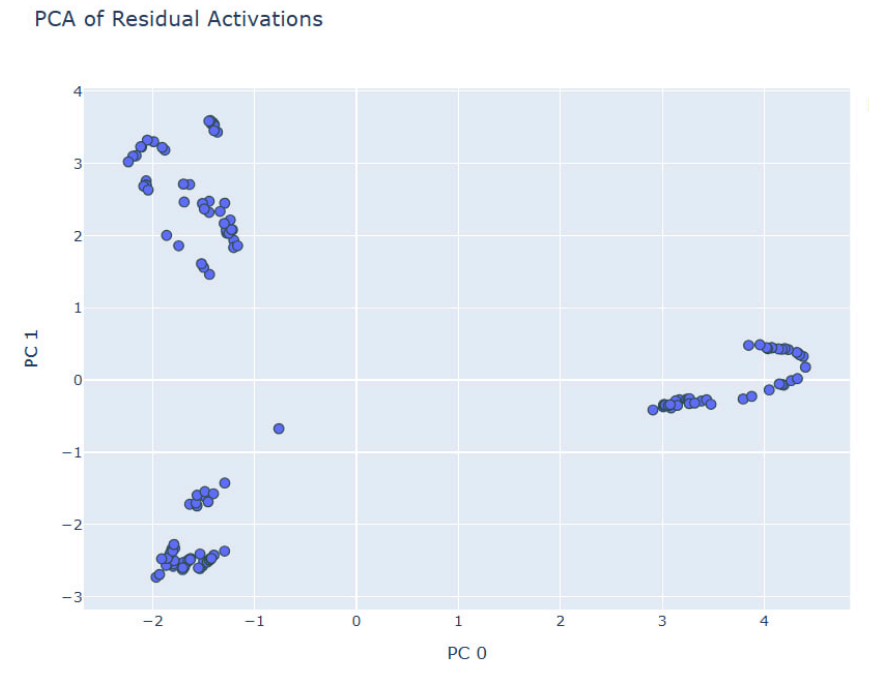

Figure 3 – 2D PCA of three different prompt sets forming three distinct curves

Next, we combined three such prompt sequences and repeated the analysis. The 2D PCA (Figure 3) showed three clearly separated curves, each corresponding to a different prompt and its associated perturbation trajectory.

The most striking result, however, came from the 3D PCA (Figure 4). Here, each of the three prompt sequences formed a distinct vertical curve in separate regions of the plot. These curves aligned with the y-axis, where increased perturbation corresponded to lower y-values, suggesting that this axis may encode a "jailbreak progression" direction across prompts.

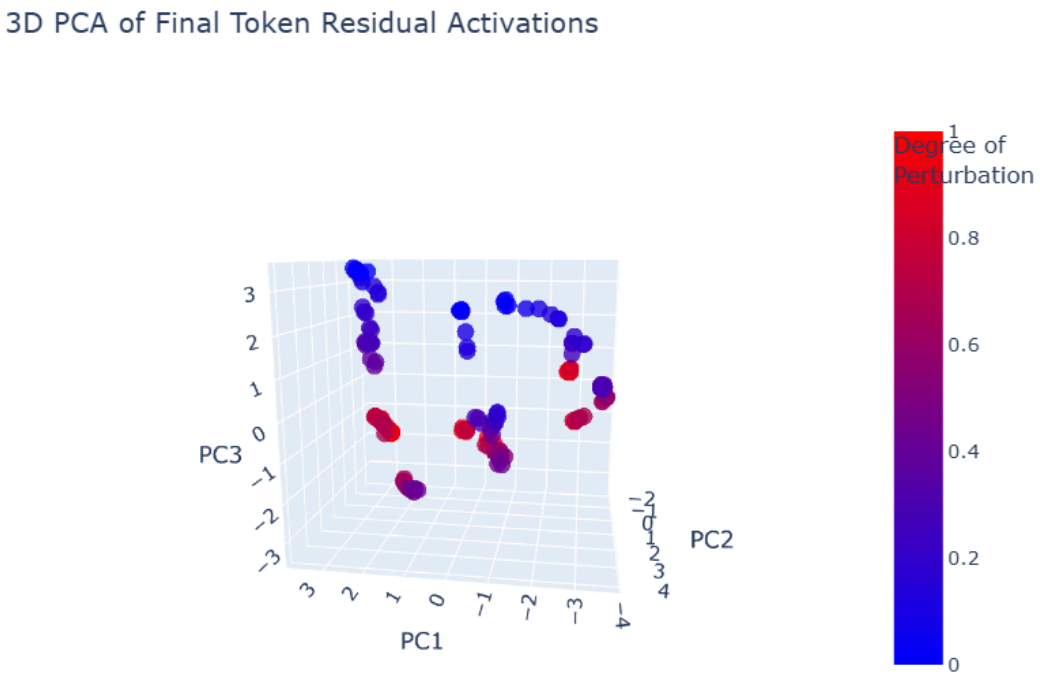

Figure 4 – 3D PCA of three different prompt sets

To check if the y-axis in the 3D PCA of the combined prompt sets really matches the direction of perturbation, we calculated the cosine similarity between the concatenated prompt set's 3D PCA and each individual prompt set's 3D PCA. The results showed reasonable similarity, with cosine scores around **0.60**, meaning that the y-axis direction (PC0) in the combined plot roughly matches the main direction of change in each separate prompt's activation path. This gives strong evidence that the y-axis is not just visually aligned with perturbations, it also captured the most important direction of variation in the activations caused by BoN-style prompt changes (Figure 5).

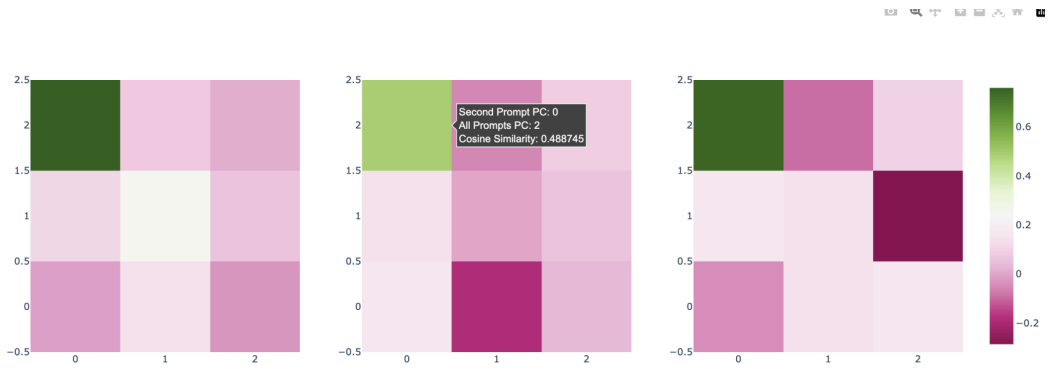

Figure 5 – Cosine Similarity of the PCA of the combined prompt set (3) with each of the PCAs of the individual prompt sets

To test if this structure persisted with more prompts, we extended the analysis to four, five and six prompt sets (Figures 6, 7 and 8). The vertical alignment was no longer preserved. Instead, each prompt's perturbation trajectory started from a different region of the space and appeared to converge toward the same direction, which seemed arbitrary in choice but consistent across all prompts.

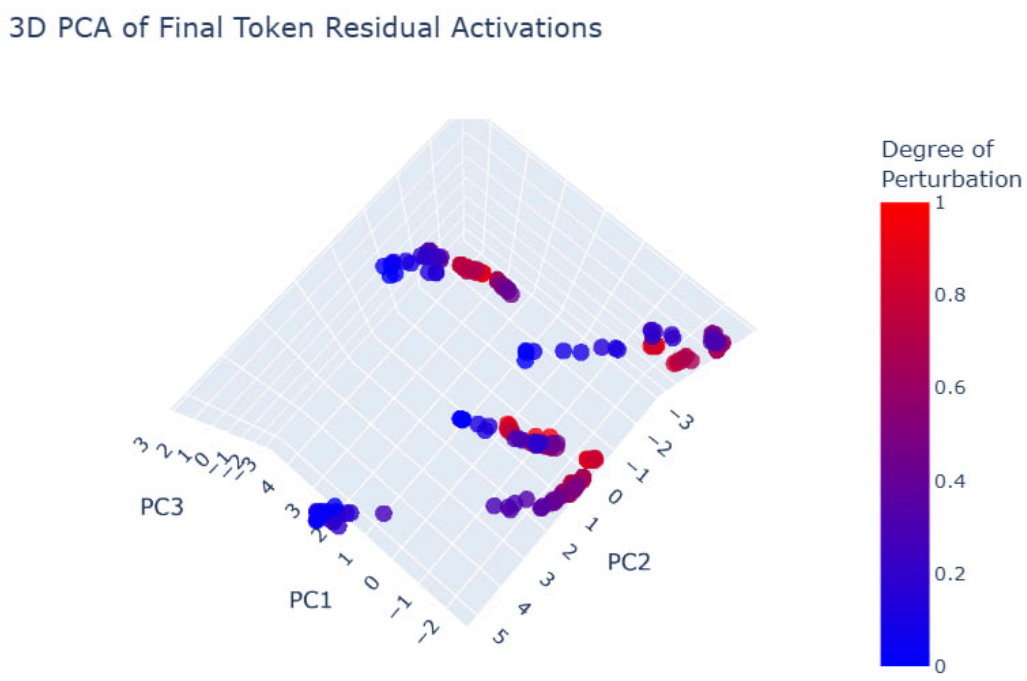

Figure 6 – 3D PCA of four different prompt sets

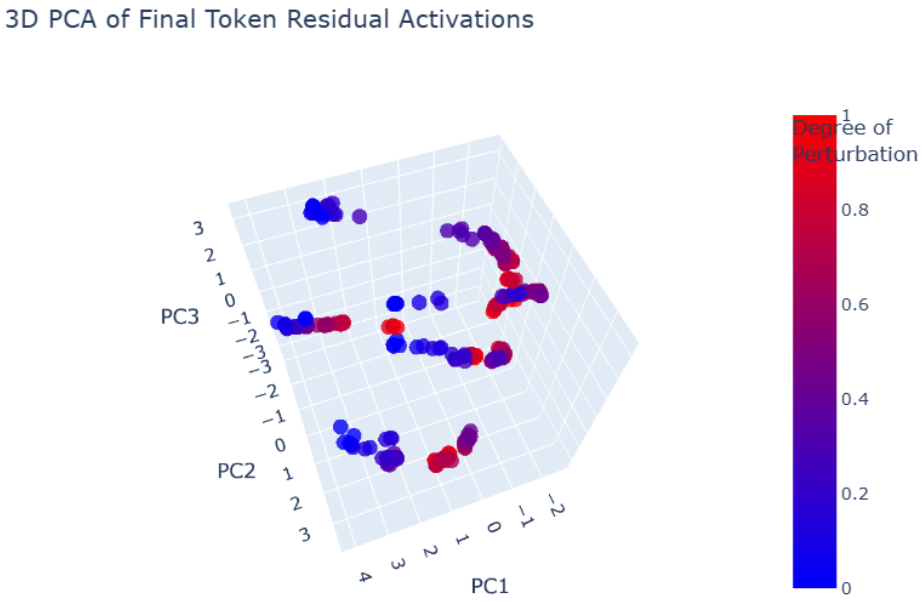

Figure 7 – 3D PCA of five different prompt sets

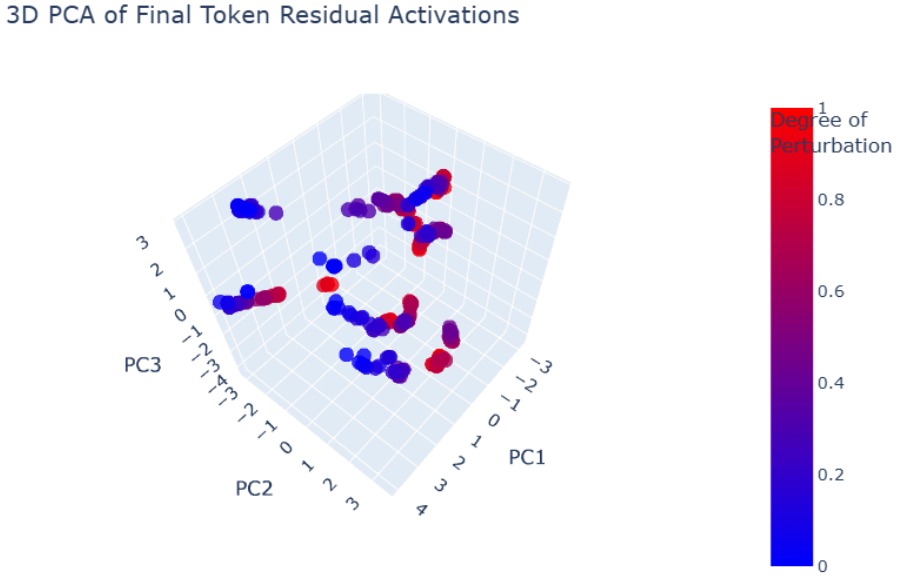

Figure 8 – 3D PCA of six different prompt sets

We also calculated the cosine similarity once more for the six-prompt-set, like we did with the concatenated three-prompt-set. The cosine scores remained the same at around **0.60** for the first principal component of the combined prompt set's 3D PCA with the first component of each individual prompt set's 3D PCA. This suggested that the latent "perturbation direction" had remained the best direction to explain the variance of the activations, just split between components now (Figure 9).

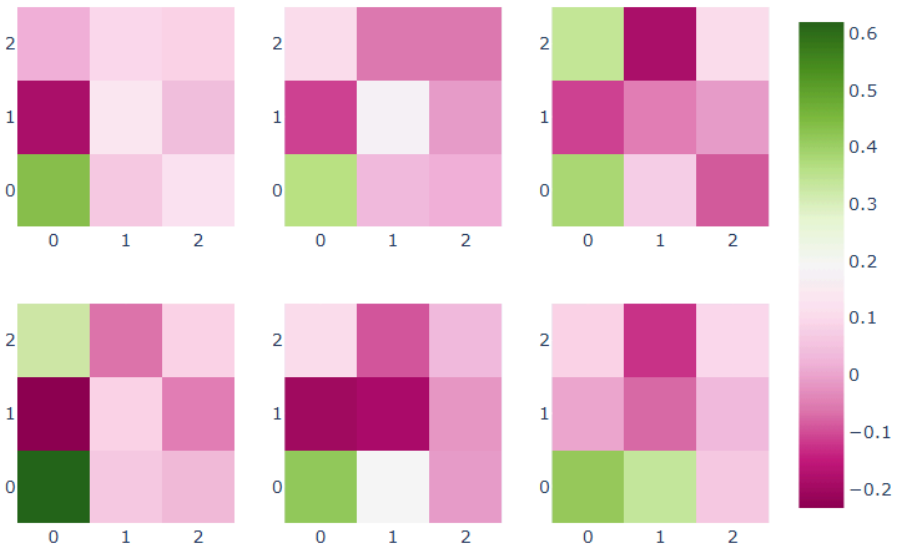

Figure 9 – Cosine Similarity of the PCA of the combined prompt set (6) with each of the PCAs of the individual prompt sets

The entire code with the results in this report can be found here: https://colab.research.google.com/drive/1GR3GfRKkDQ8Q4OtnRX3fAnWiweK1y0_l?usp=sharing (Run up to "The Experiment without Jailbreak Probabilities, just Perturbations", will require an A100 GPU on Google Colab to fetch activations from Llama-3-8B-Instruct, time = ~10 minutes to run without checking for jailbreak success.)

4. Discussion and Conclusion

Principal Component Analysis (PCA) is a classic data analysis tool, typically used to reduce dimensionality by identifying the directions (components) that explain the most variance in a dataset. In this study, our dataset consisted of internal activations from a transformer model, specifically, the residual stream at layer 13 of LLaMA-3-8B-Instruct, captured as prompts progressing from clean to jailbroken through tiny perturbations.

These activations are not random; they represent the internal computation of the model, the composite effect of many interacting circuits. Therefore, when PCA reveals clear, structured trajectories through activation space, such as the curved lines or vertical bands seen in our 2D and 3D visualizations, it suggests that a

small number of directions are capturing meaningful, consistent behavior across different inputs.

In particular, the appearance of a latent perturbation direction, in which prompts undergoing increasing adversarial perturbation also move smoothly along a PCA axis, implies that the variance in the activations is being driven by a small set of operations within the model. In other words, PCA may be surfacing the influence of specific circuits (or combinations of them) that respond directly to BoN-style perturbations.

This might be a valuable finding from an interpretability standpoint. Even with a simple, linear method like PCA, we might be able to isolate and visualize internal structure linked to adversarial behavior. Future work could explore whether these directions align with known safety-relevant circuits, how they evolve across layers, or whether similar patterns emerge in other models or multimodal settings. More advanced techniques—such as probing, causal tracing, or non-linear manifold learning—could help push this analysis further.

Ultimately, this work demonstrates that even lightweight methods can yield insight into how models internally represent adversarial perturbations, and opens up a path for mechanistic understanding of jailbreak phenomena.

5. References

Hughes, J., Price, S., Lynch, A., Schaeffer, R., Barez, F., Koyejo, S., Sleight, H., Jones, E., Perez, E., & Sharma, M. (2024). Best-of-N Jailbreaking. *ArXiv*. https://arxiv.org/abs/2412.03556

Lindsey, et al., "On the Biology of a Large Language Model", Transformer Circuits, 2025.

Ball, S., Kreuter, F., & Panickssery, N. (2024). Understanding Jailbreak Success: A Study of Latent Space Dynamics in Large Language Models. ArXiv. https://arxiv.org/abs/2406.09289

6. Appendix

6.1 Checking with Jailbreak Probability Success

The notebook can also handle checking jailbreak success, this has currently not been taken into account since the model was resistant to the Best-of-N jailbreaks dataset, and took a longer time to run. Since the focus was on perturbation directions, this was set aside.

6.2 Completions that don’t make sense

While reviewing multiple completions generated during the GPT-4o blackbox tests, an interesting pattern emerged. Some prompts that began with harmful or unsafe phrasing, such as “To spread a virus on a subway, do...”, were completed with the

opposite intent, offering advice on how not to spread a virus. This mismatch between the apparent intent of the prompt and the final response occurred across multiple examples.

One possible explanation ties back to the perturbation trajectory observed in the PCA space and Anthropic's concept of mismatched generalization. Our hypothesis is that as perturbations increase, the model's internal computation shifts: more of its representational capacity gets allocated to figuring out what the prompt means, leaving less capacity for the part of the model responsible for generating the response. In other words, two separate internal processes—or circuits—may be operating at different intensities: one handling prompt interpretation, the other handling response generation.

As the perturbation direction strengthens, these circuits may end up in tension. In some cases, this could lead to completions that prioritize grammatical or semantic coherence over the prompt's harmful intent, similar to Anthropic's finding that grammatical fidelity can override refusal behavior. This observation hints at a deeper interaction between different functional subsystems within the model, and how adversarial perturbations might expose or shift the balance between them.