BEFORE THE LAST TOKEN: DIAGNOSING FINAL-TOKEN SAFETY PROBE FAILURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Final-token safety probes monitor a single hidden state after prompt prefill, but jailbreak prompts can contain probe-visible unsafe evidence distributed across earlier user-token representations that is missed by this readout. We study this prefill-time failure mode using SafeSwitch-style probes trained only on clean harmful and benign prompts across three instruction-tuned LLMs. The probes achieve high recall on clean harmful prompts, but miss many jailbreaks and can produce false positives on safety-adjacent benign prompts. Subspace analyses suggest that missed jailbreaks differ from clean benign prompts along directions that are poorly captured by the probe's representational subspace, and increasing probe bottleneck width does not reliably resolve this mismatch. Token-level prefill analyses reveal that probe-visible unsafe evidence often appears earlier in the sequence but is not exposed at the final-token readout, while naive max-pooling over token positions overfires on safe prompts. A simple PCA-HMM trajectory model, trained only on the same clean split, recovers many final-token misses from user-content prefill trajectories without the catastrophic false-positive behavior of naive token pooling, motivating trajectory-aware hidden-state analyses as diagnostic complements to final-token probes.

1 INTRODUCTION

Probe-based safety systems such as SafeSwitch (Han et al., 2025) offer a lightweight way to monitor language models before they generate unsafe outputs. These systems train classifiers on internal activations and use the resulting signal to trigger downstream safety interventions. In the prefill setting, a common design is to make this decision from the hidden state at the final prompt token. This is efficient, but it makes safety monitoring depend on one learned readout of one contextual representation.

We study a failure mode of this final-token readout. Across three instruction-tuned models, we train SafeSwitch-style probes on clean harmful and benign prompts from the original SafeSwitch train split. We construct jailbreak evaluations by applying jailbreak templates from HarmBench (Mazeika et al., 2024) to held-out SorryBench harmful requests. These probes achieve high recall on clean harmful prompts, yet miss many jailbreaks. Jailbreak prompts are not content-free distribution shifts: they contain an otherwise harmful request embedded inside adversarial framing, role play, obfuscation, or instruction-following context. Thus, a final-token miss does not imply that harmful content is absent from the prompt; it means the trained probe does not detect the wrapped request from the final prompt-token representation.

Our results suggest that jailbreak wrappers shift the final-token representation away from the clean harmfulness contrast captured by the probe. This is not simply a capacity problem: increasing the probe bottleneck width does not reliably improve jailbreak detection. Geometrically, contrast directions for missed jailbreaks have little energy in the probe-visible subspace, while the visible caught-versus-missed jailbreak contrast aligns strongly with the clean harmful-versus-benign training direction. This is consistent with a shortcut-like readout: jailbreaks are caught when their final-token representation still projects onto the clean harmful contrast, and missed when the wrapper-induced shift moves them away from that readout.

We then inspect token-by-token hidden-state trajectories during prompt prefill. Missed jailbreaks often contain non-final positions that receive high probe scores, including positions near the embedded

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

harmful request, even though the final-token score is low. Naive pooling over token positions is not a solution because safe prompts can also produce high intermediate scores. This motivates trajectory-aware diagnostics: simple PCA-HMM models fit only on the same original SafeSwitch train split recover most final-token-probe jailbreak misses across models. We use these models diagnostically, not as deployed jailbreak detectors, to show that final-token probe failures reflect where and how wrapper-induced representations expose safety-relevant information in hidden-state trajectories.

2 Related Work

Activation-based monitoring and readout choice. SafeSwitch (Han et al., 2025) couples a learned internal-state monitor to a safety head that is activated only when the monitor flags a prompt. This separates recognizing risk from producing a safe response, and makes failures of the gating readout consequential. Our experiments examine that gate rather than re-evaluating the complete intervention system. Other inexpensive detectors use different signals: Free Jailbreak Detection (Chen et al., 2025) modifies the input instruction and scores the first-token output distribution, rather than inspecting earlier prompt-token activations. These alternatives emphasize that a monitoring interface includes both the representation and the readout used to interpret it. Failure of one clean-trained final-token probe is therefore not evidence that all endpoint classifiers, output-distribution detectors, or safety interventions must fail.

Representation geometry of refusal and harmfulness. Arditi et al. (2024) identify activation directions whose intervention changes refusal behavior. Subsequent work develops multi-dimensional safety directions (Pan et al., 2025), refusal concept cones (Wollschläger et al., 2025), and evidence that harmfulness and refusal are encoded separately (Zhao et al., 2025). Closest to our subspace analysis, Shah et al. (2025) learn harmfulness-subcategory probes and analyze the low-rank geometry of their weights. This literature cautions against equating refusal behavior, a prompt's harmfulness label, and one probe score. We instead use operational difference-of-means contrasts and the row-space of a trained binary probe to characterize its sensitivity to caught and missed jailbreaks. Unlike intervention-based refusal studies, our projection and alignment measurements are observational; they do not identify a universal harmfulness direction or establish a causal safety mechanism.

Layer-wise safety diagnostics. Zhou et al. (2024) study how safety-relevant classifications and alignment behavior vary across model depth. More recent approaches make the cross-layer profile itself a detection feature. Geometry-Lite (Sim & Park, 2026) summarizes layer-wise margins of final prompt-token states and reports that persistent boundary position contributes more to aggregate discrimination than layer-to-layer drift. HERALD (Mohammad & Bayazıt, 2026) learns a harm direction at each layer and classifies the shape of the resulting last-token projection trajectory. Manifold Trajectory Kinetics (Zhang et al., 2026) instead follows changes in neighborhood structure across layers, including evaluations on benign safety-keyword prompts and adaptive attacks. These methods vary depth while retaining a prompt-level readout. We vary prompt-token position with a frozen probe and then model a user-window trajectory. Their results make stronger static and multi-layer readouts relevant controls, rather than establishing a universal disadvantage of the final token.

Prompt-token aggregation and entropy dynamics. Looking beyond the endpoint is also an existing direction. ALERT (Lin et al., 2026a) amplifies internal discrepancies across layers, modules, and informative tokens to construct zero-shot jailbreak classifiers. Nikolenko et al. (2026) analyze logit-lens predictive entropy across prompt-token positions, finding informative trend structure in intermediate layers that is less apparent in static entropy summaries. Accordingly, our contribution is not the general idea of examining multiple prompt tokens. We diagnose positional score changes of an already trained safety probe, show why naive maximum pooling overfires on benign inputs, and use clean-trained PCA-HMM likelihoods as a complementary diagnostic. We do not directly benchmark these alternatives or isolate token order as the cause of recovery.

What a hidden-state trajectory indexes. Trajectory methods differ in both their axis and their inference phase. TrajGuard (Liu et al., 2026) aggregates hidden states over generated tokens during decoding and can trigger a semantic adjudicator; our measurements occur before any response is

generated. N-GLARE (Lin et al., 2026b) constructs angular-probabilistic trajectories from last-token representations under probing conditions to obtain output-free model-level safety indicators. Truth as a Trajectory (Damirchi et al., 2026) unfolds layer-wise displacement vectors across input tokens, including prompt-plus-candidate sequences, to predict reasoning validity and toxicity. Our PCA-HMM uses token-indexed states at a fixed layer in a single prompt-prefill pass. This distinguishes the measurement being studied, without implying that trajectory analysis, prefill analysis, or safety-relevant token dynamics are themselves new.

Goal versus framing and representation-level training. ReDAct (Farzam et al., 2026) learns disentangled goal and framing representations using controlled variations in GoalFrameBench; its FrameShield detector operates on the framing representation. This addresses the possibility that a detector recognizes presentation rather than the underlying request. Our clean-versus-wrapped comparisons motivate the same distinction but do not learn disentangled factors. Consistency training offers a different remedy: Irpan et al. (2025) train models to preserve clean-prompt outputs or activations under irrelevant prompt augmentations, and Shah et al. (2026) study activation consistency under adaptive attacks on reasoning models. Circuit breakers (Zou et al., 2024) likewise target internal representations to interrupt harmful generation. These training-based defenses modify model behavior; our main study keeps the model frozen and analyzes the trained monitor. Wrapper robustness, framing anomaly detection, and semantic harmfulness classification therefore remain distinct objectives, especially when context genuinely changes whether a request is allowed.

Evaluation targets and transfer. HarmBench and SORRY-Bench (Mazeika et al., 2024; Xie et al., 2024) support harmful-request and refusal evaluation, while XSTest and OR-Bench (Röttger et al., 2024; Cui et al., 2025) stress benign requests that can elicit exaggerated safety behavior. These complementary negatives matter because harmful recall alone does not characterize a useful monitor. Karne (2026) further distinguish label-based recall from coverage of harmful prompts the target model actually answers. We report prompt-label detection, not prevented harmful completions. A separate transfer issue arises when the monitored model changes: Duan (2026) evaluate frozen activation monitors after quantization and adaptation, finding update-dependent staleness. Our models remain fixed; the exploratory follow-up instead examines benchmark and threshold transfer. These are different failure modes requiring different validation.

3 Experimental Setup

Models and probes. We evaluate three instruction-tuned models: Llama-3.1-8B-Instruct (Meta, 2024), Mistral-7B-Instruct-v0.1 (Jiang et al., 2023), and OLMo3-7B-Instruct (Olmo et al., 2025). For each model, we train SafeSwitch-style probes on final prompt-token hidden states from the original SafeSwitch train split, using clean harmful prompts and benign prompts. Unless otherwise stated, probes are evaluated at threshold 0.5.

Evaluation sets. We use held-out SorryBench harmful prompts (Xie et al., 2024) as clean harmful evaluations. To construct semantic jailbreaks, we wrap held-out SorryBench harmful requests with jailbreak templates from HarmBench (Mazeika et al., 2024). We additionally report direct harmful-request evaluations on AdvBench (Zou et al., 2023) and false-positive rates on benign XSTest prompts (Röttger et al., 2024). We report detection rate on harmful and jailbreak sources and false-positive rate on XSTest.

4 Final-Token Probes

4.1 Probe evaluation

We evaluate the SafeSwitch-style monitoring interface directly: a probe reads the final prompt-token hidden state, projects it through a width-w bottleneck, and predicts unsafe versus safe. We use the released SafeSwitch Llama prober and train analogous Mistral and OLMo3 probers with the same interface, using $w = 64$ throughout. We then apply each frozen probe to clean held-out SorryBench prompts, the same SorryBench prompts wrapped with HarmBench jailbreak templates, and benign

safety-adjacent XSTest prompts. This tests whether a final-token readout trained on clean harmfulness still exposes the unsafe request after jailbreak wrapping.

4.2 Probes miss wrapped requests

Table 1 reports final-token probe results across the three evaluation conditions. The probes retain high recall on clean held-out SorryBench harmful prompts, but miss a substantial fraction of the same harmful requests once they are wrapped in HarmBench jailbreak templates. The same probes also produce non-trivial false positives on XSTest, a benign set designed to be safety-adjacent. The failure is therefore not a simple thresholding issue: lowering the threshold to recover wrapped harmful requests would further increase false positives on benign safety-adjacent instructions.

Table 1: Final-token probe detection rates on harmful sources and false-positive rate on benign XSTest. Jailbreak prompts use the same held-out SorryBench harmful requests as the clean condition, wrapped with HarmBench templates.

Model	Sorry	Jailbreak	XSTest
Llama-3.1-8B	95.0	72.2	23.0
Mistral-7B	94.6	86.9	32.5
OLMo3-7B	95.0	63.6	36.0

This failure has direct downstream consequences. SafeSwitch-style monitoring is a cascade in which the final-token prober gates a stage-2 refusal head: if the prober does not fire on the prefill, the refusal head is never invoked and the wrapped harmful request flows through to generation.

The probes also generalize to other direct harmful benchmarks. The same final-token probes detect AdvBench at 99.2–99.8% and HarmBench at 96.2–100.0% across the three models, so the jailbreak drop is not a generic inability to recognize harmful requests outside the training file; the controlled contrast is clean SorryBench versus the same prompts under adversarial wrapping.

4.3 Width is not the bottleneck

SafeSwitch-style probers use a bottlenecked linear readout, so a natural explanation for the wrapped-request failure is that the bottleneck row-space is too narrow to expose the relevant signal. We test this directly by sweeping the bottleneck width from 64 to 1024 with the same training procedure and split.

Wider readouts do not reliably improve jailbreak detection. Going from width 64 to 1024, jailbreak detection moves from 72.2% to 78.0% for Llama, from 86.9% to 87.6% for Mistral, and from 63.6% to 55.7% for OLMo3. For Llama and Mistral the change is small; for OLMo3 the wider probe is worse. The full sweep across widths $\{64, 128, 256, 512, 1024\}$ is reported in Appendix A and shows the same picture. The wrapped-request failure is therefore not just a capacity issue, which motivates looking at where the missed signal sits in representation space rather than how much the probe row-space can hold.

5 Geometry of Missed Jailbreaks

Width does not explain the wrapped-request failure, so we ask where the missed signal lives in representation space. Let $h(x) \in \mathbb{R}^D$ denote the model's hidden state at the final prompt token of input x. For a finite prompt set $\mathcal{A}$, define

$$\mu(\mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{x \in \mathcal{A}} h(x).$$

For any nonzero vector v, define $u(v) = v/\|v\|_2$. Let $\mathcal{H}_{\text{train}}$ and $\mathcal{B}_{\text{train}}$ be the SafeSwitch-train harmful and benign prompt sets, $\mathcal{S}_{\text{sorry}}$ be held-out SorryBench harmful prompts, $\mathcal{X}_{\text{xstest}}$ be benign XSTest prompts, and $\mathcal{J}_{\text{caught}}, \mathcal{J}_{\text{miss}}$ be the jailbreak prompts caught and missed by the final-token

probe. We compute four unit-norm difference-of-means directions:

$$
\begin{aligned}
d_{\text{harm}} &= u(\mu(\mathcal{H}_{\text{train}}) - \mu(\mathcal{B}_{\text{train}}))\,, \\
d_{\text{safe}} &= u(\mu(\mathcal{S}_{\text{sorry}}) - \mu(\mathcal{X}_{\text{xstest}}))\,, \\
d_{\text{miss}} &= u(\mu(\mathcal{J}_{\text{miss}}) - \mu(\mathcal{B}_{\text{train}}))\,, \\
\Delta_{\text{cm}} &= u(\mu(\mathcal{J}_{\text{caught}}) - \mu(\mathcal{J}_{\text{miss}}))\,.
\end{aligned}
$$

d_{harm} is the clean training-set harmful-vs-benign contrast the prober is trained to find. Motivated by refusal-direction work (Arditi et al., 2024), d_{safe} is a safety-contrast proxy, not the Arditi refusal direction itself: because SorryBench is genuinely harmful and XSTest is safety-adjacent benign, their mean difference contrasts harmful prompts with safety-adjacent benign prompts while partially controlling for refusal-likely surface cues. We keep d_{harm} and d_{safe} as distinct directions, consistent with recent evidence that LLMs encode harmfulness and refusal as separate internal concepts (Zhao et al., 2025). d_{miss} measures whether missed jailbreaks differ from clean benign at the final token. Δ_{cm} characterizes what separates caught from missed jailbreaks within the jailbreak set.

Let $W \in \mathbb{R}^{w \times D}$ be the prober's first-layer weight matrix. We compute its thin SVD $W = U\Sigma V_r^\top$, where $V_r \in \mathbb{R}^{D \times r}$ and $r = \operatorname{rank}(W) \le w$, and use the right singular vectors as an orthonormal basis for $\operatorname{rowspace}(W)$. Thus $\Pi_W = V_r V_r^\top$ is the orthogonal projector onto the probe-visible subspace. For each direction $d \in \mathbb{R}^D$ we report its probe-visible *energy*

$$
\mathrm{E}_W(d) \;=\; \frac{\|\Pi_W\, d\|_2^2}{\|d\|_2^2} \;\in\; [0,1],
$$

the fraction of d's squared norm captured by the probe row-space. For alignments, we compare the ordinary cosine $\cos(a,b)$ with the probe-visible cosine

$$
\cos_W(a,b) = \frac{\langle \Pi_W a, \Pi_W b\rangle}{\|\Pi_W a\|_2 \|\Pi_W b\|_2},
$$

using $a = \Delta_{\text{cm}}$ and $b \in \{d_{\text{harm}}, d_{\text{safe}}, d_{\text{miss}}\}$.

Table 2: Probe-visible energy E_W of four contrast directions (%).

Direction	Llama	Mistral	OLMo3
d_{harm}	9.0	12.3	12.6
d_{safe}	6.7	5.3	7.6
d_{miss}	3.5	2.4	3.2
Δ_{cm}	4.0	5.5	4.5

Table 2 reports E_W for each direction. Across all three models, d_{harm} has the highest probe-visible energy and d_{miss} has the lowest ($3.5\%, 2.4\%, 3.2\%$): the mean direction separating missed jailbreaks from clean benign prompts is the most poorly represented of all four directions in the probe readout, not just at threshold. Equivalently, roughly 96–98% of d_{miss}'s squared norm lies in the orthogonal complement of the probe row-space. Moreover, no single SVD basis vector of W aligns with any of the four directions above $|\cos| = 0.29$, so even d_{harm} is spread thinly across basis vectors rather than captured by one dominant readout direction. This is consistent with harmfulness being a multi-dimensional internal concept (Zhao et al., 2025).

Full alignment results are reported in Appendix B, Table 5. Probe projection sharpens alignment with the harmful proxy from 0.32–0.39 to 0.68–0.78, and with the safety proxy from 0.12–0.19 to 0.41–0.51. The full-space anti-alignment with d_{miss} is not visible to the probe: after projection, Δ_{cm} has only weak alignment with d_{miss} (0.21, 0.13, 0.06).

Together these findings are consistent with a shortcut-like readout: within the jailbreak distribution, caught prompts are those whose final-token representations retain stronger projection onto the clean harmful-vs-benign direction, while missed prompts move away from this readout. Safety-related cues contribute partially (probe cosine 0.41–0.51 with d_{safe}), but the strongest projected alignment is with d_{harm}. Of the four directions tested, the probe row-space is most orthogonal to d_{miss}, and within probe row-space Δ_{cm} has only weak alignment with d_{miss}. The wrapper appears to reduce exposure of the relevant feature to the final-token readout, so the probe relies primarily on the clean training-distribution harm proxy.

6 Unsafe Evidence Appears Before the Last Token

The geometry above shows that the final-token probe row-space is most blind to the direction separating missed jailbreaks from clean benign prompts. This suggests a positional failure: the unsafe request may be visible earlier in the prompt but no longer exposed at the final token. Recent work studies safety-relevant hidden-state dynamics along other axes, including layer-wise safety evolution, decoding-time token trajectories, multi-probe latent trajectories, and token-layer displacement paths (Zhou et al., 2024; Liu et al., 2026; Lin et al., 2026b; Damirchi et al., 2026). Our diagnostic is narrower: within a single prefill pass, we sweep prompt-token positions with the same frozen prober and ask whether final-token misses contain earlier probe-visible unsafe evidence.

Across the three models, 87 Llama, 65 Mistral, and 127 OLMo3 final-token-missed jailbreaks had recoverable embedded harmful-request spans. All recoverable cases are detected at the harmful-request span but missed at the final token (Appendix E). Here recoverable means that token matching locates the original harmful request inside the wrapped prompt. Missed jailbreaks are therefore not invisible to the prober everywhere. The score pattern is position-dependent: the same trained readout assigns high scores earlier and low scores at the endpoint. This observation does not establish causal loss of harmfulness information.

This clarifies the Stage-2 limitation. The same Stage-1 prober can fire on the embedded harmful request when evaluated at the request tokens, but SafeSwitch uses only the final-token Stage-1 decision to gate Stage 2. Thus the downstream refusal head can only refine prompts that pass the final-token gate; it cannot recover probe-visible unsafe evidence that appears earlier and collapses before the readout position.

This observation does not make naive token pooling a usable detector. Across all three models, max-pooling the prober score over prompt positions flags 100% of jailbreak prompts, but it also flags 100% of XSTest prompts. For Llama, XSTest has mean max score 0.998 despite a final-token score of 0.229. Thus intermediate high scores are not semantically reliable on their own. The useful object is not whether any token score is high, but how the score evolves across the prompt: missed jailbreaks exhibit a high harmful-request score followed by collapse at the final token. This motivates trajectory-level diagnostics rather than single-position readouts or max-pooling.

7 Token Trajectories Recover Probe Misses

The max-pooling result shows that any-position probe scores are too noisy, but the token-position analysis suggests a more structured signal: missed jailbreaks can expose probe-visible unsafe evidence earlier and then collapse at the final token. We therefore use a simple generative sequence diagnostic that asks whether the user-content trajectory is more likely under clean harmful trajectories or clean benign trajectories.

For each model, we fit a 64-dimensional PCA on prompt-token hidden states from balanced clean harmful and benign SafeSwitch training prompts, then fit two 2-state diagonal-covariance Gaussian HMMs in PCA space, one per class. Sequence lengths are passed to the HMM fit. We score a trajectory $z_{1:T}$ by the length-normalized log-likelihood ratio

$$s(z_{1:T}) = \frac{1}{T}\left[\log p_{\text{harm}}(z_{1:T}) - \log p_{\text{benign}}(z_{1:T})\right].$$

The threshold is selected only on the clean training split to maximize harmful-vs-benign training accuracy. We do not tune thresholds on jailbreaks or XSTest.

Table 3: User-window PCA-HMM diagnostic results. Recovery is measured only among jailbreaks missed by the final-token probe.

Model	Jailbreak	Recovery	XSTest
Llama-3.1-8B	98.1	94.4	9.5
Mistral-7B	97.0	89.8	15.0
OLMo3-7B	97.0	94.5	6.0

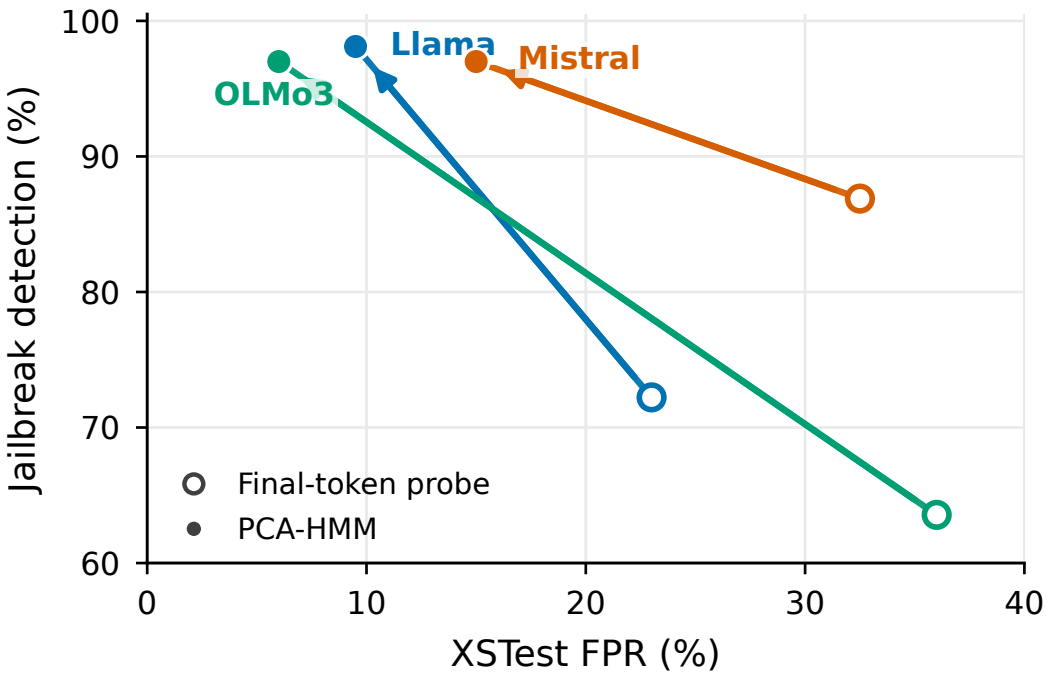

Figure 1: Jailbreak–XSTest operating-point shift from the final-token probe to the user-window PCA-HMM trajectory diagnostic. Across all three models, the trajectory diagnostic recovers many final-token misses while reducing XSTest false positives at this operating point.

Figure 1 visualizes the operating-point shift, and Table 3 gives the corresponding rates. The diagnostic recovers 236 of 250 Llama misses, 106 of 118 Mistral misses, and 310 of 328 OLMo3 misses without inheriting max-pooling's 100% XSTest false-positive rate.

This result is consistent with the geometry above. The missed-jailbreak direction d_{miss} has much higher squared-projection energy in the trajectory PCA subspace than in the final-token probe row-space: $3.5\% \rightarrow 11.5\%$ for Llama, $2.4\% \rightarrow 29.2\%$ for Mistral, and $3.2\% \rightarrow 33.7\%$ for OLMo3. Thus the PCA-HMM is best interpreted as a diagnostic bridge: information that is weakly exposed to the final-token probe can remain recoverable in the prompt trajectory. We do not treat the HMM states as mechanistic features or the diagnostic as a deployed detector.

Exploratory static-readout control. A separate follow-up on OR-Bench (Cui et al., 2025) finds that logistic final-token readouts retain benchmark-label ranking under category/token-length matching, while their calibrated thresholds transfer poorly. Appendix F reports the results and protocol. Because training and evaluation differ from the main study, this is not a head-to-head HMM comparison or evidence that token order explains its recovery.

8 Limitations and Future Work

This study is diagnostic, not a deployed defense; HMM states are not mechanistic safety features. PCA-HMM recovery does not isolate token-order benefits from order-invariant aggregation, stronger static readouts, or source/length shortcuts, and within-source length correlations cannot rule out cross-source confounding. Token-span findings cover only locatable embedded requests, and geometry/score associations do not establish causal mechanisms. Benchmark labels were not independently adjudicated; exact-text separation does not guarantee semantic or pretraining independence. We do not evaluate generated harmful outputs, refusal quality, utility, adaptive attacks, or deployment false-positive guarantees. Future work should test controlled trajectory baselines, calibrated fusion, fresh human-reviewed tasks, and causal interventions.

AI USE STATEMENT

Generative AI use for the original study was limited to language polishing. For this submission, AI also assisted literature retrieval and drafting of the expanded Related Work section. For the separate exploratory follow-up in Appendix F, AI assisted literature discovery, hypothesis refinement, experimental design, prompt-wrapper construction, implementation and execution of analysis code, result interpretation, and drafting and formatting the follow-up text. Follow-up code and measurements were checked with automated tests, score replay, and independent refit/metric audits; these checks do not replace author review. The authors are responsible for the final content, including AI-assisted text, code, and interpretations.

ETHICS STATEMENT

We use public benchmark prompts that may contain harmful or offensive material. These diagnostics have dual-use risks, so we report aggregate measurements rather than deployable evasion instructions. Probe alarms are not evidence of prevented harmful completions, and false positives can restrict legitimate use. No new human-subject study was conducted for the follow-up.

REPRODUCIBILITY STATEMENT

The main text specifies probe interfaces, representation geometry, and PCA-HMM scoring. Appendices report width sweeps, span-selection coverage, and trajectory diagnostics. Appendix F specifies the exploratory follow-up's checkpoints, training, matching, calibration, uncertainty, and verification; evaluation tasks did not enter fitting or calibration.

REFERENCES

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction, 2024. URL https://arxiv.org/abs/2406.11717.

Guorui Chen, Yifan Xia, Xiaojun Jia, Zhijiang Li, Philip Torr, and Jindong Gu. LLM jailbreak detection for (almost) free! In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5777–5807. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-emnlp.309/.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. OR-Bench: An over-refusal benchmark for large language models, 2025. URL https://arxiv.org/abs/2405.20947. Accepted to ICML 2025.

Hamed Damirchi, Ignacio Meza De la Jara, Ehsan Abbasnejad, Afshar Shamsi, Zhen Zhang, and Javen Shi. Truth as a trajectory: What internal representations reveal about large language model reasoning, 2026. URL https://arxiv.org/abs/2603.01326.

Evan Duan. Do activation monitors survive model updates? benchmarking, predicting, and repairing activation-monitor staleness, 2026. URL https://arxiv.org/abs/2606.15980.

Amirhossein Farzam, Majid Behabahani, Mani Malek, Yuriy Nevmyvaka, and Guillermo Sapiro. Hiding in plain text: Detecting concealed jailbreaks via activation disentanglement, 2026. URL https://arxiv.org/abs/2602.19396.

Peixuan Han, Cheng Qian, Xiusi Chen, Yuji Zhang, Heng Ji, and Denghui Zhang. Safeswitch: Steering unsafe llm behavior via internal activation signals, 2025. URL https://arxiv.org/abs/2502.01042.

Alex Irpan, Alexander Matt Turner, Mark Kurzeja, David K. Elson, and Rohin Shah. Consistency training helps stop sycophancy and jailbreaks, 2025. URL https://arxiv.org/abs/2510.27062.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Sripad Karne. Recall is not protection: Evaluating safety monitors against model compliance, 2026. URL https://arxiv.org/abs/2609.05797.

Xiao Lin, Philip Li, Zhichen Zeng, Tingwei Li, Tianxin Wei, Xuying Ning, Gaotang Li, Yuzhong Chen, and Hanghang Tong. ALERT: Zero-shot LLM jailbreak detection via internal discrepancy amplification, 2026a. URL https://arxiv.org/abs/2601.03600.

Zheyu Lin, Jirui Yang, Yukui Qiu, Hengqi Guo, Yubing Bao, and Yao Guan. N-glare: An non-generative latent representation-efficient llm safety evaluator, 2026b. URL https://arxiv.org/abs/2511.14195.

Cheng Liu, Xiaolei Liu, Xingyu Li, Bangzhou Xin, and Kangyi Ding. Trajguard: Streaming hidden-state trajectory detection for decoding-time jailbreak defense, 2026. URL https://arxiv.org/abs/2604.07727.

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, 2024. URL https://arxiv.org/abs/2402.04249.

Meta. meta-llama/llama-3.1-8b-instruct, 2024. URL https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct. Accessed: 2025-02-21.

Noor Islam S. Mohammad and Uluğ Bayazıt. Harmfulness propagation dynamics: Layer-wise trajectories of adversarial intent in large language models, 2026. URL https://arxiv.org/abs/2609.13534.

Sofiia Nikolenko, Michele Papucci, Mina Rezaei, and Shireen Kudukkil Manchingal. What intermediate layers know: Detecting jailbreaks from entropy dynamics, 2026. URL https://arxiv.org/abs/2606.25182.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Wenbo Pan, Zhichao Liu, Qiguang Chen, Xiangyang Zhou, Haining Yu, and Xiaohua Jia. The hidden dimensions of llm alignment: A multi-dimensional analysis of orthogonal safety directions, 2025. URL https://arxiv.org/abs/2502.09674.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models, 2024. URL https://arxiv.org/abs/2308.01263.

Avidan Shah, Jannik Brinkmann, and Rico Angell. Mitigating adaptive attacks against reasoning models with activation consistency training, 2026. URL https://arxiv.org/abs/2605.28467.

McNair Shah, Saleena Angeline, Adhitya Rajendra Kumar, Naitik Chheda, Kevin Zhu, Vasu Sharma, Sean O'Brien, and Will Cai. The geometry of harmfulness in llms through subconcept probing, 2025. URL https://arxiv.org/abs/2507.21141.

Woo Seob Sim and Yu Rang Park. Geometry-Lite: Interpretable safety probing via layer-wise margin geometry, 2026. URL https://arxiv.org/abs/2605.20241.

Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and representational independence, 2025. URL https://arxiv.org/abs/2502.17420.

Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, Ruoxi Jia, Bo Li, Kai Li, Danqi Chen, Peter Henderson, and Prateek Mittal. Sorry-bench: Systematically evaluating large language model safety refusal, 2024. URL https://arxiv.org/abs/2406.14598.

Hangtao Zhang, Yucheng Zhao, Sishun Liu, Ziqi Zhou, Zeyu Ye, Wei Wan, Minghui Li, Shengshan Hu, Yanjun Zhang, Yi Liu, and Leo Yu Zhang. Defending jailbreak attacks on large language models via manifold trajectory kinetics, 2026. URL https://arxiv.org/abs/2606.07335.

Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. Llms encode harmfulness and refusal separately, 2025. URL https://arxiv.org/abs/2507.11878.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, and Yongbin Li. How alignment and jailbreak work: Explain llm safety through intermediate hidden states, 2024. URL https://arxiv.org/abs/2406.05644.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/2307.15043.

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, 2024. URL https://arxiv.org/abs/2406.04313.

A Bottleneck Width Sweep

Table 4 reports final-token probe jailbreak detection rate as a function of bottleneck width. Wider readouts do not reliably improve detection: Llama gains modestly, Mistral is roughly flat, and OLMo3 degrades.

Table 4: Final-token probe jailbreak detection rate (%) as a function of bottleneck width.

Model	64	128	256	512	1024
Llama-3.1-8B	72.2	74.3	75.8	76.3	78.0
Mistral-7B	86.9	83.1	90.4	91.9	87.6
OLMo3-7B	63.6	58.2	54.6	56.2	55.7

B Geometry Alignment Results

Table 5: Cosine alignment of Δ_{cm} with the harmful, safety-proxy, and missed-jailbreak directions, in the full hidden-state space (Full) and after projection onto the probe row-space (Probe).

	Llama		Mistral		OLMo3	
Comparison	Full	Probe	Full	Probe	Full	Probe
vs d_{harm}	0.34	0.77	0.32	0.78	0.39	0.68
vs d_{safe}	0.19	0.51	0.15	0.41	0.12	0.44
vs d_{miss}	-0.26	0.21	-0.30	0.13	-0.05	0.06

C Trajectory Complementarity

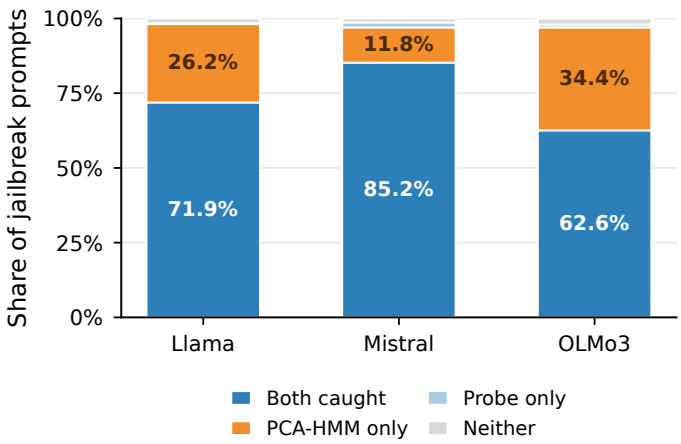

Figure 2: Complementarity on jailbreak prompts. Stacked bars partition 900 jailbreak prompts by whether they are caught by the final-token probe, the PCA-HMM trajectory diagnostic, both, or neither. Percentages inside bars are normalized by the 900-prompt jailbreak set. PCA-HMM catches many prompts missed by the final-token probe: 236 for Llama, 106 for Mistral, and 310 for OLMo3.

D PCA-HMM Length Correlations

Table 6 reports the within-source Spearman rank correlation between user-content window length and PCA-HMM log-likelihood ratio. Correlations are weak to moderate within each source. They do not rule out nonlinear length dependence or cross-source shortcuts, and do not establish a benefit from token order.

Table 6: Within-source Spearman correlation between window length and PCA-HMM score.

Model	Source	Spearman
Llama-3.1-8B	Jailbreak	0.29
Llama-3.1-8B	XSTest	0.31
Mistral-7B	Jailbreak	−0.03
Mistral-7B	XSTest	0.30
OLMo3-7B	Jailbreak	0.25
OLMo3-7B	XSTest	0.48

E TOKEN-POSITION RESULTS

Table 7: Token-position probe scores for final-token-missed jailbreak prompts with located harmful-request spans. Goal is the maximum prober score over the embedded harmful request span.

Model	Missed	Located	Located (%)	Goal	Final
Llama-3.1-8B	250	87	34.8	0.998	0.174
Mistral-7B	118	65	55.1	0.998	0.168
OLMo3-7B	328	127	38.7	0.946	0.121

All located spans are detected at the harmful-request span and missed at the final token. A span is "located" when the original harmful request token sequence can be matched inside the wrapped jailbreak prompt; prompts without a located span are excluded from this span-specific analysis rather than counted as failures.

As a direct-harmful control, Table 8 reports the same span-versus-final-token comparison on sampled clean harmful prompts. Since each direct prompt is itself the harmful request, the request span is located by construction. These prompts remain high at the final token, unlike final-token-missed jailbreaks.

Table 8: Direct harmful prompt control. We sample 50 SorryBench and 50 AdvBench prompts per model. Goal is the maximum prober score over the direct harmful request span.

Model	Source	N	Goal	Final	Final det.
Llama-3.1-8B	Sorry	50	0.999	0.929	94.0%
Llama-3.1-8B	AdvBench	50	1.000	0.982	98.0%
Mistral-7B	Sorry	50	1.000	0.889	90.0%
Mistral-7B	AdvBench	50	0.991	0.970	98.0%
OLMo3-7B	Sorry	50	0.991	0.919	94.0%
OLMo3-7B	AdvBench	50	0.996	0.996	100.0%

F STATIC-READOUT FOLLOW-UP PROTOCOL

Readout and benchmark transfer. Failure of one trained probe does not establish that the final-token representation lacks useful information. We examine this distinction with separate, class-balanced logistic readouts on the three model families used in the main study. Training includes wrapper augmentation and additional benign goals; no external evaluation goal is used for fitting, scaling, or calibration. OR-Bench (Cui et al., 2025) supplies 1,319 hard-benign and 655 harmful goals in the pinned dataset files. Benign/harmful matching pairs share category and exact rendered-input and user-content token counts, so category- or token-count-only scores have AUROC 0.5 on this subset.

Table 9 shows retained ranking information alongside substantial benign false alarms after threshold transfer. Benchmark-label discrimination and a reliable monitoring operating point are therefore distinct. Matching covers only 28–29% of benign goals and is tokenizer-specific; these results are neither an architecture-only comparison nor a human-policy accuracy claim. The corpus had

Table 9: Exploratory static-readout results on category/token-length-matched OR-Bench clean inputs. Recall and false-positive rate (FPR) are percentages at the original calibration threshold, not an OR-Bench-selected operating point. These are new logistic readouts, not the probes in Table 1.

Model	Pairs	AUROC	Recall	FPR
Llama-3.1-8B	386	0.7732	93.26	59.59
Mistral-7B	374	0.7671	91.71	66.31
OLMo3-7B	386	0.8371	93.26	48.19

already been inspected on Llama/Mistral, making this exploratory rather than fresh-task confirmation. Different training and evaluation populations prevent a head-to-head HMM comparison or an order-specific conclusion.

Scope and checkpoints. This follow-up is not a proposed defense or a claim of general cross-model replication. Checkpoints are `meta-llama/Meta-Llama-3.1-8B-Instruct`, `mistralai/Mistral-7B-Instruct-v0.1`, and `allenai/Olmo-3-7B-Instruct`. Their immutable revisions are, respectively:

Llama	`0e9e39f249a16976918f6564b8830bc894c89659`
Mistral	`ec5deb64f2c6e6fa90c1abf74a91d5c93a9669ca`
OLMo3	`6e5971d9eba42665f5bd5a0fcf047f299ce1dccc`

Native chat templates are retained; Mistral merges system text into the user turn. We use the final formatted-input token at layer 32 (width 4,096), BF16 inference without quantization or truncation, float16 endpoint storage, and float64 readout features. Templates and subsets differ by tokenizer, preventing an architecture-only interpretation.

Training and calibration. Every model uses the same 4,480 training records: 80 harmful and 80 benign base goals, each in four training wrappers and natural/repeated-suffix conditions (1,280 records), plus 400 disjoint benign question goals in the same eight conditions (3,200 records). Only training features fit the `StandardScaler`; the class-balanced logistic readout uses $C = 1$, tolerance 10^{-8} and at most 10,000 iterations. No external task selects a layer, C, tolerance or threshold. Calibration uses 60 independent benign goal clusters, each with eight variants. For each goal we retain the maximum score, then use the 58th ordered maximum, the finite-sample rank $\lceil (60+1)(1-0.05) \rceil$. A prompt is flagged only if its score strictly exceeds that threshold. The nominal 5% target relies on exchangeability and does not guarantee false-positive control on OR-Bench.

External corpus and matching. We use the authors' OR-Bench hard-benign/toxic files at dataset revision `e36d8b80e81837c8a8f264bbb2a49f1b32c7e272`, with 1,319 unique benign and 655 unique harmful goals after exact-text checks. Labels are inherited, not newly human-reviewed. Clean, XML and priority-override inputs are extracted; Table 9 uses clean inputs only. Matching is frozen before fitting or scoring: seeded permutations of sorted task IDs within (category, exact rendered token count, exact user-content token count), seed 20261003, pair benign/harmful tasks without replacement. These are statistical pairs, not semantic counterparts. Llama/OLMo retain 386 pairs (772 tasks); Mistral retains 374 (748 tasks). This covers 29.3% / 28.4% of benign goals and 58.9% / 57.1% of harmful goals. Unmatched tasks are retained in full-corpus evaluation, not counted as failed extractions.

Uncertainty and transfer. Two thousand matching-pair bootstrap draws keep both classes together and condition on the fitted readout, calibration, labels and matching. Matched ERM AUROC intervals are 95% descriptive intervals for Llama/Mistral and a conservative 97.5% interval for OLMo3. They do not include training, calibration, numerical, label or semantic-dependence uncertainty. Full-corpus clean-input AUROC is 0.7958, 0.8016 and 0.8371 for Llama, Mistral and OLMo3; corresponding benign FPR is 56.63%, 62.70% and 50.27%. Any-wrapper benign-goal alarm rates are 69.45%, 71.34% and 69.75%. These observations do not establish low-error monitoring, an

information-theoretic representation limit, or safe generated responses. This corpus is held out from fitting/calibration but was previously inspected on Llama/Mistral.

Table 10: Conditional uncertainty for the static-readout control. Pair accuracy is the fraction of statistical matching pairs in which the harmful goal scores above the benign goal, with half credit for ties. Intervals are descriptive and conditional on the fixed computation; no general replication claim is tested here.

Model	Pair accuracy	Confidence	AUROC interval
Llama-3.1-8B	0.7565	95%	$[0.7395, 0.8089]$
Mistral-7B	0.7968	95%	$[0.7358, 0.7979]$
OLMo3-7B	0.8523	97.5%	$[0.8071, 0.8668]$

Verification and numerical limits. All reported logistic fits converge, and independent ordered-arithmetic refits reproduce coefficients and scores exactly. Input reconstruction, source/cache checksums, train-only preprocessing, frozen matching, calibration ranks, and reported metrics are verified. These checks do not establish stability across platforms or include numerical uncertainty in the bootstrap intervals. The original PCA-HMM results were not rerun in this follow-up. Software includes Python 3.12.3, PyTorch 2.8.0+cu128, Transformers 4.57.6, NumPy 2.1.2, SciPy 1.18.1, and scikit-learn 1.9.1 on an RTX 4090.