BEYOND MECHANISMS: AN EVENT-LEVEL CHARACTERIZATION OF LLM JAILBREAKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work has uncovered multiple internal mechanisms underlying jailbreaks, including changes in harmfulness recognition, refusal, and so on. Mechanism-level perspectives, however, are inherently fragmented: different studies typically isolate different factors or processes involved in jailbreaks, often at different levels of abstraction. As a result, they provide important local explanations of jailbreak behavior, but do not by themselves characterize how these factors jointly change when a jailbreak succeeds. We instead shift the unit of analysis from mechanism to event, asking which underlying factors, taken together, characterize the successful jailbreak event. Because reduced refusal does not necessarily imply substantive assistance toward the harmful goal, we treat harmful compliance as a separate outcome factor. To this end, we first investigate how harmfulness recognition, refusal, and harmful compliance jointly change during a successful jailbreak event, and find a consistent pattern across three LLM families: harmfulness recognition remains largely unchanged, while refusal decreases and harmful compliance increases. This event-level finding naturally motivates us to formulate jailbreak suppression as a constrained problem: increasing refusal and reducing harmful compliance under the constraint that harmfulness recognition remains approximately unchanged. We then implement this formulation with a constrained restoration method. Experiments across three LLM families show that the resulting method consistently reduces jailbreak success.

1 INTRODUCTION

Prompt-level studies. Jailbreaks expose a fundamental failure mode of safety-aligned large language models (LLMs), in which carefully constructed prompts can cause a model to provide harmful assistance that it would otherwise refuse. Early research has largely developed a prompt-level understanding of jailbreaks, focusing on how changes in prompt form and context affect whether safety alignment is bypassed, and how successful jailbreaks are reflected in model outputs. This line of work has identified a wide range of prompt manipulation and attack strategies, ranging from manually designed jailbreak prompts to automatically optimized adversarial prompts (Wei et al., 2023; Zou et al., 2023), together with evaluation protocols for determining whether the resulting responses provide substantive assistance toward harmful goals (Souly et al., 2024; Mazeika et al., 2024; Chao et al., 2024). Collectively, these studies have established a broad prompt-level understanding of how jailbreaks are elicited and manifested in model behavior.

Mechanism-level studies. Beyond prompt-level studies, recent work has increasingly turned to the internal mechanisms underlying jailbreaks. Rather than asking which forms of prompts can elicit unsafe behavior, these studies ask what changes inside the model, e.g., typically model representations, when a jailbreak succeeds. Existing work has revealed mechanisms associated with several safety-related factors, including harmfulness recognition, refusal, and so on (Arditi et al., 2024; Zhao et al., 2026; Wu et al., 2026; Zhao et al., 2025; Han et al., 2025; Yin et al., 2026; He et al., 2026). These studies show that harmfulness recognition can persist when refusal weakens, and that compliance-related behavior may involve distinct internal processes. Recognizing a harmful request, refusing it, and proceeding toward unsafe assistance therefore need not depend on a single mechanism. This work has shifted jailbreak research from observable prompt-response patterns toward explaining how safety-related processes operate inside models.

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.

The Event-level Jailbreak Hypothesis

When the same harmful goal shifts from direct refusal to successful jailbreak, harmfulness recognition remains approximately unchanged, while refusal decreases and harmful compliance increases.

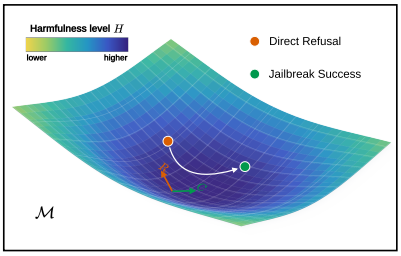

Figure 1: Event-level view of jailbreaks. For the same harmful goal, we compare the model state under a refused direct harmful request with that under a successful jailbreak. The arrow denotes the change associated with the jailbreak event, and the background indicates the harmfulness level. H, R, and C denote harmfulness recognition, refusal, and harmful compliance, respectively.

Challenges. Despite this progress, mechanism-level studies tend to focus on specific changes during successful jailbreaks, such as shifts in harmfulness recognition, refusal, or other internal processes. These analyses reveal important parts of the phenomenon, but they do not by themselves characterize the jailbreak event as a whole. A successful jailbreak may involve coordinated changes in multiple safety-relevant factors, and studying these factors separately does not reveal how they change together when jailbreak success is actually realized. This distinction is particularly important for refusal: weakening or removing refusal is not sufficient to establish jailbreak success, because a non-refusal response need not provide substantive assistance toward the harmful objective. Conversely, a model may show safety or refusal-related cues while providing harmful assistance. If we treat a successful jailbreak as a single event, it remains unclear which safety-relevant factors change together and which remain approximately unchanged.

Contributions: Event-level characterization. To address the limitation above, we move beyond individual mechanisms and focus on the successful jailbreak event [1] itself. Specifically, we ask how the relevant safety-related factors change together when the same harmful goal shifts from direct refusal to successful jailbreak. To this end, we characterize this event along three factors repeatedly implicated by prior mechanism-level studies: harmfulness recognition, refusal, and harmful compliance. As illustrated in Figure 1, we observe a consistent event-level pattern: harmfulness recognition remains largely unchanged, while refusal decreases and harmful compliance increases. This finding addresses the fragmentation above by bringing these previously studied factors into a single characterization of jailbreak success. It also naturally suggests how jailbreaks may be suppressed: increasing refusal and reducing harmful compliance while keeping harmfulness recognition approximately unchanged. In summary, our contributions are as follows:

- Instead of analyzing individual mechanisms in isolation, as in prior work, we introduce an event-level perspective on jailbreak success that takes the jailbreak event as the unit of analysis and jointly characterizes multiple safety-related factors and how they change together.
- By comparing direct refusal and successful jailbreak for the same harmful goal, we identify a consistent event-level signature: harmfulness recognition remains largely unchanged, while refusal decreases and harmful compliance increases (Sec. 2).
- Building on this characterization, we formulate jailbreak suppression as increasing refusal and reducing harmful compliance while preserving harmfulness recognition, and develop a concept learning method that follows this principle and consistently reduces jailbreak success (Sec. 3).
- Experiments across three LLM families show that the proposed method consistently reduces jailbreak success while keeping harmfulness recognition approximately unchanged (Sec. 4).

2 Event-Level Characterization of Jailbreak Success

In this section, we study jailbreak success from an event-level perspective. We first introduce the same-goal comparison between direct refusal and successful jailbreak in Sec. 2.1. We then examine

[1] Here, "event-level" denotes the goal-conditioned transition from a refused direct request to a successful jailbreak for the same harmful objective. Multiple successful attack prompts for the same goal are treated as realizations of one transition, not as individual events.

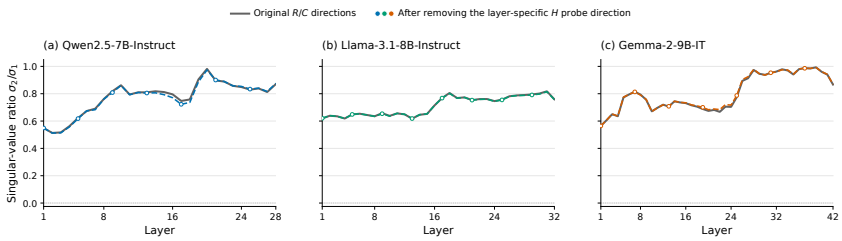

Figure 2: Layerwise refusal–compliance geometry across three models. Solid curves show the singular-value ratio σ_2/σ_1 of the unit-normalized R/C probe directions; dashed curves show the same quantity after projecting out the layer-specific H direction. R/C remains non-collinear across layers, with little change after removing H. The final point is measured after final normalization.

why refusal alone is insufficient to characterize jailbreak success in Sec. 2.2. Finally, we investigate how harmfulness recognition, refusal, and harmful compliance change jointly in Sec. 2.3.

2.1 Setup and Notation: An Event-Level View of Jailbreak Success

We view a successful jailbreak event as the transition from direct refusal to successful jailbreak for the same harmful goal. Specifically, for each harmful goal g, let x_g^{DHR} denote a direct harmful request whose response satisfies the refusal criterion. Let $x_{g,a}^{\mathrm{JS}}$, $a \in \{1, \ldots, N_g\}$, denote the a-th successful jailbreak variant of the same harmful goal, where N_g is the number of successful jailbreak variants available for goal g. Each $x_{g,a}^{\mathrm{JS}}$ produces a response that provides harmful assistance and is therefore judged as a successful jailbreak. The event of interest is the change from direct refusal to successful jailbreak while keeping the underlying harmful goal fixed.

Based on this setup, we ask what changes when a jailbreak succeeds. The transition from direct refusal to successful jailbreak involves three questions: whether the model still recognizes the request as harmful, whether it continues to refuse the request, and whether it provides harmful assistance toward the requested goal despite containing refusal. Accordingly, we consider three factors: harmfulness recognition (H), refusal (R), and harmful compliance (C). Harmfulness recognition and refusal capture whether the model recognizes the request as harmful and whether it rejects the request, respectively, as also highlighted by prior work (Zhao et al., 2026; Wu et al., 2026). Harmful compliance captures whether the response actually provides assistance toward the harmful goal.

Our event-level view differs from mechanism-level analyses in the unit of analysis. A successful jailbreak is a complete event that may involve coordinated changes in multiple safety-relevant factors. Existing mechanism-level studies typically isolate particular changes within this event, such as changes in harmfulness recognition, refusal, or other internal mechanisms. While such analyses reveal important parts of the process, they do not by themselves characterize how these factors jointly change when jailbreak success is realized. Instead, our event-level perspective takes jailbreak success as a whole and seeks to characterize the joint changes of multiple safety-relevant factors.

2.2 Is Harmful Compliance Needed Beyond Refusal?

The event-level view aims to characterize jailbreak success jointly through H, R, and C, as mentioned above. For this characterization to be meaningful, these factors should provide complementary information instead of simply restating one another. Prior work has already shown that harmfulness recognition H and refusal R can dissociate (Zhao et al., 2026; Wu et al., 2026; He et al., 2026). Refusal captures refusal- or safety-related behavior, whereas harmful compliance captures substantive assistance toward the harmful goal. Reduced refusal does not necessarily imply harmful assistance, and refusal-related cues may coexist with such assistance. The remaining question is whether C contributes information beyond refusal. If C were simply the opposite of refusal, then C would add no information beyond R. We thus examine whether R and C capture distinct information, before turning to their joint changes with H.

We first examine this question at the response level using model responses generated from public JailbreakBench (Chao et al., 2024) prompts. Let $y_R, y_C \in \{0, 1\}$ denote the independently assigned

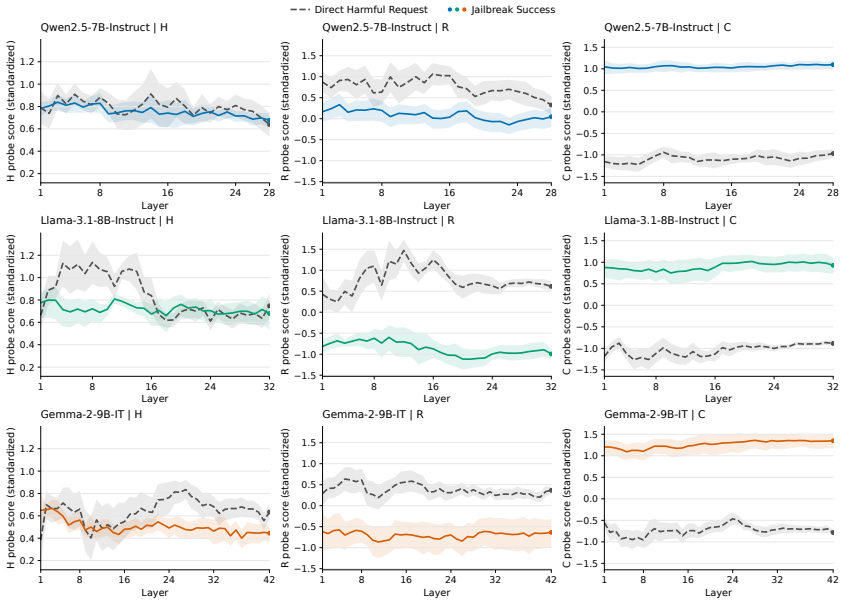

Figure 3: **Layerwise evidence for the event-level jailbreak hypothesis.** Rows show different models; columns show H, R, and C. Gray dashed and colored solid curves denote goal-matched DHR and JS states, respectively; shading shows pointwise 95% goal-bootstrap confidence intervals.

refusal and harmful-compliance labels. If the two were strict complements, every response would fall into either $(y_R, y_C) = (1, 0)$ or $(0, 1)$. However, 23.1%–44.6% of the responses violate this relation across the three models. Most violations occur at $(y_R, y_C) = (1, 1)$, where a response contains refusal, policy, or safety-framing cues while still providing harmful assistance. Thus, refusal alone does not determine whether a response actually assists the harmful goal. Equivalently, conditioning on the same refusal/safety-cue label does not uniquely determine the harmful-assistance outcome, so replacing C by $1 - R$ would discard response-semantic distinctions present in the data. Further details on response generation and labeling are provided in Appendix B.1.

We further test whether the two factors collapse to opposite directions in representation space. At each layer, we separately fit linear probes for refusal and harmful compliance from representations. We use the resulting probe directions to characterize their geometry, following prior work on concept-related directions in LLM representations (Li et al., 2023; Marks & Tegmark, 2023; Tigges et al., 2023). Specifically, we compute the singular-value ratio σ_2/σ_1 of the matrix formed by the two unit-normalized probe directions. A ratio of zero indicates collinearity, whereas one indicates orthogonality. Figure 2 shows that the R/C probe directions remain non-collinear across layers in all three models. Projecting out the layer-specific harmfulness direction and renormalizing the residual directions produces little change in these ratios. Further details are provided in Appendix B.5.

Together, response- and representation-level results show that harmful compliance contains information beyond refusal, and their probe directions do not reduce to a single linear axis.

2.3 The Event-Level Jailbreak Hypothesis

Having established that C preserves response-outcome information not determined by R alone, we now examine how H, R, and C change jointly within the same goal-matched jailbreak transition. To examine this question, we compare direct refusal and successful jailbreak for the same harmful goal using standardized probe scores for the concepts.

Let $h_\ell(x)$ denote the pre-generation, prompt-boundary representation of prompt x at layer ℓ. For each $q \in \{H, R, C\}$, a separately trained linear probe maps this representation to a decision logit $r_{q,\ell}(x)$. We define the corresponding standardized score as

$$s_{q,\ell}(x) = \frac{r_{q,\ell}(x) - m_{q,\ell}}{\tau_{q,\ell}}, \tag{1}$$

where $m_{q,\ell}$ and $\tau_{q,\ell}$ are the mean and standard deviation of the probe logits on the probe-fitting examples [2]. All probe parameters and normalization statistics are frozen before test evaluation and shared across the DHR and JS conditions. Details on probe fitting and held-out AUROCs of the $H/R/C$ probes across all evaluated layers are reported in Appendix B.2.3.

We then characterize the change associated with a jailbreak event for each $q \in \{H, R, C\}$ as

$$\Delta q(g) = \frac{1}{N_g} \sum_{a=1}^{N_g} \left[s_q(x_{g,a}^{\mathrm{JS}}) - s_q(x_g^{\mathrm{DHR}}) \right]. \tag{2}$$

Thus, $\Delta q(g)$ measures the average change in factor q from direct refusal to successful jailbreak for harmful goal g. We then average $\Delta q(g)$ across harmful goals so that each goal contributes equally, and denote the resulting goal-balanced mean by Δq.

As shown in Figure 3, the event-level pattern is broadly observed across model depth in all three models. Harmfulness recognition remains approximately unchanged at most evaluated layers, with pointwise 95% confidence intervals for the mean changes contained within the prespecified equivalence region. In contrast, refusal scores consistently decrease and harmful-compliance scores consistently increase at every evaluated layer.

These results show that successful jailbreaks can occur without a substantial loss of harmfulness recognition. For the same harmful goal, harmfulness recognition remains approximately unchanged, while refusal decreases and harmful compliance increases:

$$\boxed{\Delta H \approx 0, \qquad \Delta R < 0, \qquad \Delta C > 0.} \tag{3}$$

This event-level pattern directly motivates the suppression objective developed in the next section.

3 Jailbreak Suppression by the Event-Level Characterization

Section 2 shows that successful jailbreaks are characterized by approximately unchanged harmfulness recognition, decreased refusal, and increased harmful compliance, as summarized in Eq. 3. This suggests a simple principle for jailbreak suppression:increasing refusal and reducing harmful compliance while preserving harmfulness recognition. Under this formulation, we use harmfulness recognition as a specificity constraint for suppression, with R and C as the response-policy targets and seek an intervention that reverses the jailbreak-associated R/C shift while keeping H approximately unchanged.

To this end, we seek a vector v in representation space whose effect reverses the jailbreak-associated changes in Eq. 3. Let $\Delta_v q$ denote the change in score s_q after applying v, relative to before applying v, for $q \in \{H, R, C\}$. The desired suppression effect is

$$\boxed{\Delta_v H \approx 0, \qquad \Delta_v R > 0, \qquad \Delta_v C < 0.} \tag{4}$$

Eq. 4 tells us what effect v should have, but not how to obtain it. To realize the first term, $\Delta_v R - \Delta_v C$, we use the paired representations of direct refusal and successful jailbreak. In particular, for the same harmful goal, moving from successful jailbreak back toward direct refusal corresponds to increasing refusal and decreasing harmful compliance. We therefore use the reverse representation change from successful jailbreak to direct refusal, i.e., $h(x_g^{\mathrm{DHR}}) - h(x_{g,a}^{\mathrm{JS}})$, as the initial direction for v, see Sec. 3.1. The event-level analysis also shows that harmfulness recognition changes only slightly on average. To preserve this property conservatively, we remove from the initial direction the representation components that are associated with harmfulness recognition, see Sec. 3.2.

3.1 Constructing the Initial Suppression Vector

For each harmful goal, we average the representations of its successful jailbreak variants and compare the result with the corresponding direct-refusal representation. We then average this difference

[2] The H score is fitted only on unwrapped harmful and benign requests and frozen before the DHR–JS analysis, so it measures whether the harmfulness signal learned from direct inputs is retained under jailbreak.

across goals:

$$v_{\text{init}} = \frac{1}{G}\sum_{g=1}^{G}\left[h(x_g^{\text{DHR}}) - \frac{1}{N_g}\sum_{a=1}^{N_g} h(x_{g,a}^{\text{JS}})\right], \tag{5}$$

where G is the number of harmful goals. Thus, v_{init} represents the average change from successful jailbreak back toward direct refusal for the same harmful goal. Applying this direction is therefore expected to increase refusal and decrease harmful compliance.

However, this representation difference may also contain changes related to harmfulness recognition. To address the constraint on $\Delta_v H$, we further remove from v_{init} the components that are strongly associated with harmfulness recognition.

3.2 Preserving Harmfulness Recognition

The initial vector v_{init} is constructed to increase refusal and decrease harmful compliance, but it may also contain changes that affect harmfulness recognition. To address the constraint on $\Delta_v H$, we identify representation components that are strongly associated with harmfulness recognition and remove the corresponding parts from v_{init}.

Specifically, we first decompose the LLM representation into a set of latent components using concept component analysis (Liu et al., 2026a). Other representation-decomposition methods, such as sparse autoencoders, could be feasible for the same purpose (Huben et al., 2024; Gao et al., 2025). This decomposition allows us to examine concept-related information at the viewpoint of the disentangled human-interpretable concept rather than only through the full representation. We then use the harmfulness score $s_H(x)$ to identify the components that are most strongly associated with harmfulness recognition. Let Q_H denote an orthonormal basis for the retained sensitivity subspace of the selected H-aligned components at the intervention layer. We stack the alignment-weighted gradients of the selected components with respect to the intervention-layer hidden state across development prompts, retain the leading right singular vectors, and project v_{init} onto the orthogonal complement of the resulting subspace:

$$\tilde{v} = \left(I - Q_H Q_H^\top\right) v_{\text{init}}. \tag{6}$$

Since this operation may reduce its magnitude, we rescale $\tilde{v}$ to match the magnitude of v_{init}:

$$v = \frac{\|v_{\text{init}}\|_2}{\|\tilde{v}\|_2}\tilde{v}. \tag{7}$$

The resulting vector v retains the change toward increased refusal and decreased harmful compliance while reducing the components associated with harmfulness recognition. The resulting vector is orthogonal to the retained component-sensitivity subspace while preserving the norm of v_{init}. We evaluate its effects on harmfulness, refusal, and harmful compliance in Section 4.

4 Experiments

We evaluate the proposed method from three aspects. First, we examine whether harmfulness recognition is better captured by multiple representation components than by a single component, which supports the component-level protection used in Sec. 3.2. Second, we evaluate whether the proposed method reduces jailbreak success while inducing the desired changes in the three factors: $\Delta_v H \approx 0$, $\Delta_v R > 0$, and $\Delta_v C < 0$. We further compare different forms of harmfulness protection and use additional controls to examine the role of the constructed suppression direction. Finally, we evaluate whether the method preserves utility on benign inputs.

4.1 Experimental Setup

Models and data. We evaluate three instruction-tuned LLMs: Qwen2.5-7B-Instruct (Qwen et al., 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Gemma-2-9B-IT (Team, 2024). Our attack set contains 858 public jailbreak prompts from five JailbreakBench (Chao et al., 2024) attack families. We also use benign prompts from WildJailbreak (Jiang et al., 2024) for utility evaluation.

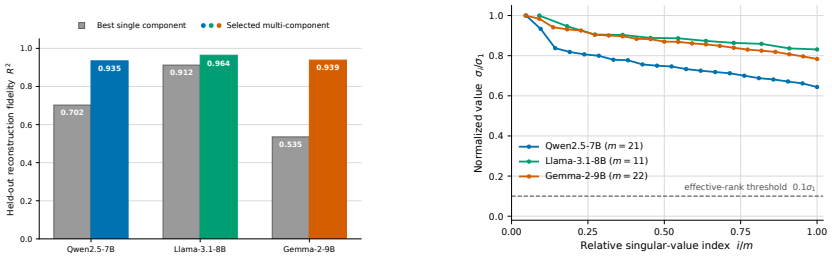

Figure 4: Multi-component recovery of the operational H component score. Left: held-out fidelity of the best single component and multi-component estimator. Right: singular-value spectra of the unit-normalized decoder directions, plotted as σ_i/σ_1 against relative spectral position i/m.

All model parameters remain frozen, and all scoring functions, component decompositions, and suppression vectors are constructed separately for each model.

Component-recovery evaluation. To evaluate whether harmfulness recognition is better captured by multiple components, we split the data by harmful-goal or benign-control identity into disjoint recovery, alignment, validation, and test partitions. All prompts associated with the same identity remain in the same partition. The component decomposition, component selection, and estimators are fixed before evaluation on the test partition. The resulting datasets contain 656, 610, and 676 samples for Qwen, Llama, and Gemma, respectively.

Suppression evaluation. The suppression test prompts are disjoint from all prompts used to construct the suppression vector, select its configuration, or determine the application gate. Because harmful goals and attack families may recur, we evaluate generalization to unseen prompts, not to unseen goals or attack families. The final test cohorts contain 532/99, 564/103, and 517/98 prompts/goals for Qwen, Llama, and Gemma, respectively. All model-specific settings are fixed before evaluation. We report attack success rate (ASR) on the full test cohort and measure changes in the frozen H, R, and C scores on prompts receiving suppression. Confidence intervals are computed by resampling harmful goals with all associated prompt variants. Data splits and evaluation settings are provided in Appendix C.1.

4.2 Recovering Harmfulness Recognition from Multiple Components

Sec. 3.2 protects harmfulness recognition at the component level. We first examine whether the harmfulness score s_H is better recovered from multiple representation components than from a single component. For each model, we compare the selected multi-component estimator with the best single component under the same procedure. Both are fixed before held-out evaluation.

As shown in Figure 4, using multiple components consistently improves recovery of s_H. The held-out R^2 increases from 0.702 to 0.935 for Qwen, from 0.912 to 0.964 for Llama, and from 0.535 to 0.939 for Gemma. The corresponding improvements are $0.232\,[0.146,\ 0.372]$, $0.052\,[0.028,\ 0.086]$, and $0.403\,[0.274,\ 0.609]$, respectively. The same result is observed across three independently trained component decompositions for each model, with positive ΔR^2 intervals in all $9/9$ runs.

The selected components span multiple directions. Within the learned decomposition, the harmfulness score is recovered more accurately by the selected component group than by the best individual component, motivating the component-level protection in Sec. 3.2.

4.3 Jailbreak Suppression

We next evaluate whether the proposed suppression method reduces jailbreak success while inducing the desired changes in harmfulness recognition, refusal, and harmful compliance.

Overall suppression performance. We first compare the initial suppression vector v_{init} with two variants that protect harmfulness recognition. The first removes the single linear direction associated with H, while the second uses the component protection introduced in Sec. 3.2. All configurations are fixed before evaluation. As shown in Figure 5(a), the initial suppression vector already reduces

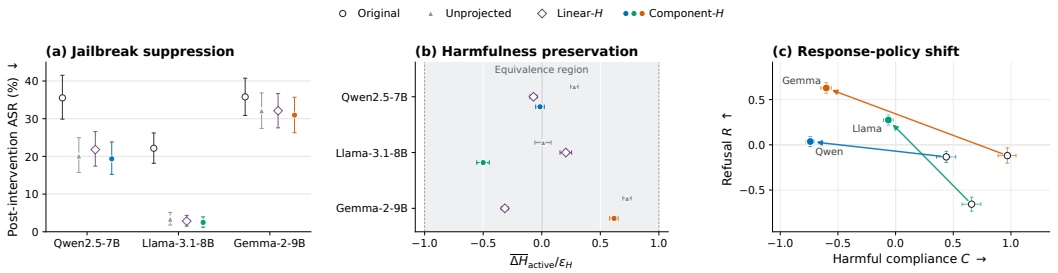

Figure 5: Jailbreak suppression results. (a) Post-suppression ASR for the original model, the initial suppression vector, single-direction H protection, and component-level H protection. (b) Mean change in harmfulness recognition on prompts to which suppression is applied, normalized by the model-specific tolerance ϵ_H. The shaded region denotes $[-1, 1]$. (c) Refusal and harmful-compliance scores before and after component-level suppression on the same prompts. Error bars show 95% confidence intervals.

Table 1: Overall jailbreak ASR (%, ↓) under different harmfulness-protection variants. Green cells indicate the lowest ASR within each model.

Model	Initial vector	Linear-H	Component-H
Qwen2.5-7B	20.11	21.80	19.36
Llama-3.1-8B	3.37	2.84	2.48
Gemma-2-9B	32.11	32.11	30.95

jailbreak success across all three models. ASR decreases from 35.53% to 20.11% for Qwen, from 22.16% to 3.37% for Llama, and from 35.78% to 32.11% for Gemma. With component-level protection, the post-suppression ASRs are further reduced to 19.36%, 2.48%, and 30.95%, respectively.

Changes in harmfulness recognition, refusal, and compliance. We then examine whether the full method produces the changes predicted by Sec. 3. On prompts to which suppression is applied, the mean change in harmfulness recognition remains within the same equivalence region used in Sec. 2, while refusal increases and harmful compliance decreases, as shown in Figure 5(b–c). Thus, the suppression method produces the opposite pattern to that observed during successful jailbreaks: $\Delta_v H \approx 0, \Delta_v R > 0, \Delta_v C < 0$.

Suppression under harmfulness constraints. We compare the unprotected initial vector, protection using the single linear H direction, and component-level protection. As shown in Table 1, all three variants keep the mean H change within the equivalence region, while component-level protection achieves the lowest post-suppression ASR for all three models. This shows that component-level protection preserves harmfulness recognition without sacrificing the suppression effect. The unprojected actuator also satisfies H-equivalence empirically, we introduce the explicit H constraint for intervention specificity, without assuming that H drift occurs in the unprojected condition. We further discuss the role of H as a preservation constraint in Appendix E.1.2.

Direction control. We further test whether the suppression effect depends on the constructed direction by comparing it with 20 projected random directions of the same magnitude. As shown in Appendix E.3, the constructed direction consistently yields substantially lower ASR and outperforms every random draw for all three models.

Semantic validation. We examine whether strict-success reversals under natural restoration yield coherent and safe responses. Such responses account for most reversals for Qwen and Llama, but only a minority for Gemma. Gemma's low safe fraction primarily reflects residual harmful compliance, showing that the mean natural displacement does not reliably restore safe responses in this model. Additional details are provided in Appendix E.3.

Targeted steering. We replace the natural displacement with a targeted response-policy direction under the same component-level H protection. This further reduces full-cohort ASR in all three models while satisfying the final-layer H-equivalence criterion on gate-active prompts. The im-

provement is largest for Gemma, whose ASR decreases from 30.95% to 10.64%, while the proportion of strict-success reversals judged coherent and safe increases from 36.59% to 95.68%. These results show that suppression effectiveness varies with the intervention even under the same H-protection framework. Detailed results are reported in Appendix E.4.

Comparison with direct refusal steering. A natural alternative is to suppress jailbreaks by directly increasing refusal. Following Arditi et al. (2024), we construct a refusal direction from harmful and harmless examples and apply it without the harmfulness-preservation step used in our method. Direct refusal steering strongly reduces jailbreak success, reaching ASRs of 2.33%, 0.00%, and 0.00% for Qwen, Llama, and Gemma, respectively. However, this reduction comes with a substantial cost. For Qwen and Gemma, the change in harmfulness recognition exceeds the prespecified equivalence region, and all three models exhibit substantial over-refusal on benign inputs.

These results highlight an important difference between simply increasing refusal and the suppression objective considered in this work. A very low ASR can be achieved by making the model refuse much more broadly, but this does not preserve its behavior on benign inputs and, for two of the three models, does not preserve harmfulness recognition. Our objective instead seeks to reduce harmful compliance and increase refusal while keeping harmfulness recognition approximately unchanged. Additional details are provided in Appendix E.6.

4.4 Benign Utility

We evaluate whether jailbreak suppression affects behavior on plain-benign and style-only prompts using the same frozen settings. We observe no additional over-refusal and only small changes in response quality. The protocol and complete results are provided in Appendix E.5.

5 Conclusion

In this work, we study jailbreak success from an event-level perspective, focusing on how multiple safety-relevant factors change together when a jailbreak succeeds.Separating R from C is important because refusal-related behavior alone does not determine whether the model ultimately provides substantive assistance toward the harmful goal; C explicitly tracks this outcome semantics. By comparing direct refusal and successful jailbreak for the same harmful goal, we find a consistent pattern across three LLM families: harmfulness recognition remains approximately unchanged, while refusal decreases and harmful compliance increases. This finding shows that successful jailbreaks do not necessarily involve a substantial loss of harmfulness recognition. Based on this characterization, we further study jailbreak suppression by reversing the observed pattern: increasing refusal and decreasing harmful compliance while preserving harmfulness recognition. We construct a suppression direction from the reverse change between successful jailbreak and direct refusal, and further remove components strongly associated with harmfulness recognition. Experiments across three LLM families show that this approach reduces jailbreak success while keeping harmfulness recognition approximately unchanged.

6 Limitations

Our analysis has several limitations. First, H, R, and C capture three important safety-related factors, but they do not exhaust all factors that may change during a jailbreak. Second, our component analysis shows that the harmfulness score is better captured by multiple learned components than by the best individual component. This does not mean that harmfulness recognition is inherently multidimensional or that the learned decomposition is unique. Third, the matched DHR–JS comparison changes both prompt form and response outcome. It therefore characterizes the joint change associated with successful jailbreaks, but does not separately identify the contribution of each source of variation. Finally, the component-recovery evaluation uses held-out harmful-goal and benign-control identities, whereas the suppression evaluation uses unseen prompts but may contain harmful goals and attack families that also appear during development. Our suppression results therefore demonstrate that the constructed direction can reduce jailbreak success under the evaluated setting, but do not establish that this direction is necessary, unique, or captures all mechanisms underlying jailbreak success.

Ethics Statement

This work studies internal changes associated with LLM jailbreaks and develops a method for suppressing them. We use public jailbreak benchmarks for controlled evaluation and do not introduce new attack methods. The work is intended to improve understanding of LLM safety and does not involve human subjects or private personal data.

Reproducibility Statement

We provide details of the datasets, data splits, probe construction, component decomposition, suppression method, evaluation protocols, and model-specific settings in the appendix to support reproduction of our results.

AI Use Statement

Generative AI tools, including OpenAI ChatGPT, were used to assist with experimental analysis, code development, and manuscript editing. All methodological decisions, statistical analyses, and scientific conclusions were determined and verified by the authors.

GPT-4o mini was additionally used as a blinded secondary judge in the cross-adjudication experiment described in Appendix B.1.4. The authors take full responsibility for all content and results.

Bibliography

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. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Sarah Ball, Frauke Kreuter, and Nina Panickssery. Understanding jailbreak success: A study of latent space dynamics in large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 250–279, 2026.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems*, 37:55005–55029, 2024.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, volume 2025, pp. 26721–26754, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Peixuan Han, Cheng Qian, Xiusi Chen, Yuji Zhang, Denghui Zhang, and Heng Ji. Safeswitch: Steering unsafe llm behavior via internal activation signals. *arXiv preprint arXiv:2502.01042*, 2025.

Zeqing He, Zhibo Wang, Zhixuan Chu, Huiyu Xu, Wenhui Zhang, Qinglong Wang, and Rui Zheng. {JailbreakScope}: Interpreting jailbreak mechanism through representation and circuit analyses. In *35th USENIX Security Symposium (USENIX Security 26)*, 2026.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, et al. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *Advances in Neural Information Processing Systems*, 37:47094–47165, 2024.

Su-Hyeon Kim, Hyundong Jin, Yejin Lee, and Yo-Sub Han. Craft: Circuit-guided refusal feature selection via cross-layer transcoders. *arXiv preprint arXiv:2604.01604*, 2026.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

Yuhang Liu, Erdun Gao, Dong Gong, Anton van den Hengel, and Javen Qinfeng Shi. Concept component analysis: A principled approach for concept extraction in llms. *arXiv preprint arXiv:2601.20420*, 2026a.

Yuhang Liu, Dong Gong, Yichao Cai, Erdun Gao, Zhen Zhang, Biwei Huang, Mingming Gong, Anton Hengel, and Javen Qinfeng Shi. I predict therefore i am: Is next token prediction enough to learn human-interpretable concepts from data? In *International Conference on Learning Representations*, volume 2026, pp. 135613–135658, 2026b.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Hadas Orgad, Boyi Wei, Kaden Zheng, Martin Wattenberg, Peter Henderson, Seraphina Goldfarb-Tarrant, and Yonatan Belinkov. Large language models generate harmful responses using a distinct mechanism, shared across harm types. *arXiv preprint arXiv:2604.09544*, 2026.

Nirmalendu Prakash, Yeo Wei Jie, Amir Abdullah, Ranjan Satapathy, Erik Cambria, and Roy Ka-Wei Lee. Beyond i'm sorry, i can't: Dissecting large-language-model refusal. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37830–37838, 2026.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Leheng Sheng, Changshuo Shen, Weixiang Zhao, Junfeng Fang, Xiaohao Liu, Zhenkai Liang, Xiang Wang, An Zhang, and Tat-Seng Chua. Alphasteer: Learning refusal steering with principled null-space constraint. In *International Conference on Learning Representations*, volume 2026, pp. 25861–25904, 2026.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, et al. A strongreject for empty jailbreaks. *Advances in Neural Information Processing Systems*, 37:125416–125440, 2024.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in neural information processing systems*, 36:80079–80110, 2023.

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. *arXiv preprint arXiv:2502.17420*, 2025.

Jinman Wu, Yi Xie, Shen Lin, Shiqian Zhao, and Xiaofeng Chen. Knowing without acting: The disentangled geometry of safety mechanisms in large language models. *arXiv preprint arXiv:2603.05773*, 2026.

Yuxin Xiao, Sana Tonekaboni, Walter Gerych, Vinith Suriyakumar, and Marzyeh Ghassemi. When style breaks safety: Defending llms against superficial style alignment. *arXiv preprint arXiv:2506.07452*, 2025.

Wei Jie Yeo, Nirmalendu Prakash, Clement Neo, Ranjan Satapathy, Roy Ka-Wei Lee, and Erik Cambria. Understanding refusal in language models with sparse autoencoders. In *EMNLP (Findings)*, pp. 6377–6399, 2025.

Yanchen Yin, Dongqi Han, and Linghui Li. Robust harmful features under jailbreak attacks: Mechanistic evidence from attention head specialization in large language models, 2026. URL https://arxiv.org/abs/2606.28153.

Lei Yu, Virginie Do, Karen Hambardzumyan, and Nicola Cancedda. Robust llm safeguarding via refusal feature adversarial training. In *International Conference on Learning Representations*, volume 2025, pp. 5254–5277, 2025.

Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. Llms encode harmfulness and refusal separately. *Advances in Neural Information Processing Systems*, 38:140283–140318, 2026.

Weixiang Zhao, Jiahe Guo, Yulin Hu, Yang Deng, An Zhang, Xingyu Sui, Xinyang Han, Yanyan Zhao, Bing Qin, Tat-Seng Chua, et al. Adasteer: Your aligned llm is inherently an adaptive jailbreak defender. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24570–24588, 2025.

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. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2461–2488, 2024.

Sicheng Zhu, Brandon Amos, Yuandong Tian, Chuan Guo, and Ivan Evtimov. Advprefix: An objective for nuanced llm jailbreaks. *Advances in Neural Information Processing Systems*, 38: 89990–90014, 2026.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Part I

Appendix

Table of Contents

A Related Work **14**

B Shared Operational Probe Score and Goal-Matched Inference **14**
B.1 Operational Groups and Outcome Adjudication 14
B.2 Frozen Score Construction and Validation 16
B.3 Cohort construction and data roles 18
B.4 Goal-level estimation and inference 18
B.5 Additional Refusal–Compliance Diagnostics 19

C Implementation and Reproducibility Details **20**
C.1 Data Roles and Evaluation Boundaries 20
C.2 Representation Extraction and Generation 21
C.3 ConCA Training 21
C.4 Component Screening and Estimation 22
C.5 Component-Protected Natural Restoration 23

D Additional Structural Details **24**
D.1 Recovery protocol. 24
D.2 Reconstruction fidelity. 25
D.3 Dictionary-seed robustness 25
D.4 Held-out recovery. 26
D.5 Decoder geometry. 26

E Additional Functional Evaluation **27**
E.1 Functional Evaluation Protocol 27
E.2 Full-Cohort and Gate-Active Natural Restoration 28
E.3 Direction Controls and Semantic Safety 29
E.4 Targeted Steering under Multi-Component H Protection 30
E.5 Benign and Style-Only Utility 31
E.6 Refusal-Direction Control 32

A Related Work

Accounts of jailbreak success. Behavioral accounts attribute jailbreak vulnerability to competing objectives and mismatched generalization (Wei et al., 2023), while controlled studies show that semantically irrelevant style patterns can alter attack success (Xiao et al., 2025). Representation-level analyses further report shared jailbreak-induced shifts, changes in harmfulness-related representations, and disruptions in the transformation from early safety concepts to refusal tokens (Ball et al., 2026; Zhou et al., 2024).

Safety concepts and activation-space control. Refusal behavior can be causally modulated through low-dimensional activation directions (Arditi et al., 2024), and input-level attacks often move representations against such behaviorally defined refusal features (Yu et al., 2025). Because standard harmful–harmless contrasts pair input semantics with opposite output policies, however, these directions need not isolate refusal from harmfulness. Subsequent work separates the two and shows that some jailbreaks suppress refusal while preserving an internal harmfulness signal (Zhao et al., 2026); related work studies this recognition–execution dissociation across layers (Wu et al., 2026). Runtime methods use safety-related activation signals for adaptive or sparse steering (Zhao et al., 2025; Sheng et al., 2026), while SafeSwitch combines instruction-safety and compliance probes to trigger a refusal head (Han et al., 2025). Furthermore, JailbreakScope (He et al., 2026) links jailbreaks to shifts in safety representations and refusal–affirmation signals.

From directions to components. A single direction need not exhaust the internal organization of refusal: gradient-based analyses identify multiple independent refusal directions and concept cones (Wollschläger et al., 2025), while SAE and circuit studies localize multiple, interacting refusal-related features (Yeo et al., 2025; Prakash et al., 2026; Kim et al., 2026). At a finer level, attention-head analyses distinguish components suppressed by attacks from safety-aligned components whose activations persist (Yin et al., 2026), and parameter-level interventions separate harmful response generation from harmfulness recognition (Orgad et al., 2026). These studies localize safety mechanisms but do not determine how an operational final-layer harmfulness score is organized. Recent theory relates last-layer next-token representations to linear mixtures of latent-concept log-posteriors (Liu et al., 2026b). Building on this result, ConCA uses regularized linear unmixing to construct a concept dictionary (Liu et al., 2026a).

Taken together, prior work shows that harmfulness recognition, refusal, and compliance-related behavior are encoded in model representations and can be modulated independently. Their joint organization under successful jailbreaks, however, remains insufficiently understood.

B Shared Operational Probe Score and Goal-Matched Inference

This appendix defines the operational groups and frozen $H/R/C$ function scores underlying the goal-matched observational analysis. The structural and functional experiments use separate development and evaluation boundaries described in Appendices D and E.

B.1 Operational Groups and Outcome Adjudication

B.1.1 Prompt Conditions.

For each harmful goal g, x_g^{DHR} denotes its direct formulation, retained when the model refuses the request. An intent-preserving attack transformation $x_{g,a}^{\mathrm{att}}$ is designated as a successful jailbreak $x_{g,a}^{\mathrm{JS}}$ when its completion is adjudicated as substantive harmful compliance. The main analysis compares goal-matched DHR and JS states, as summarized in Table 2. Benign and style-only controls are introduced separately for the utility evaluation.

Same-goal successful–failed comparisons are sparse: only 2, 1, and 2 goals contain both outcomes within the same attack family for Qwen, Llama, and Gemma, respectively. We therefore use the refused direct harmful request as the goal-matched reference and measure the shift from direct refusal to successful jailbreak. The estimand is defined at the event level beyond the attack-instance level.

Table 2: Operational groups for the main goal-matched DHR–JS comparison.

Group	Prompt	Required outcome	Role
DHR	Direct harmful request for goal g	Refusal	Goal-matched reference
JS	Intent-matched attack transformation of the same goal g	Harmful compliance	Successful jailbreak

Table 3: Summary of layerwise hypothesis support under different StrongREJECT success thresholds. G denotes the number of matched harmful goals, and Joint reports the number of layers satisfying the joint $H/R/C$ criterion.

Model	τ	G	Joint layers
Qwen2.5-7B-Instruct	0.5	16	28/28
	0.6	16	28/28
	0.7	14	28/28
	0.8	8	24/28
Llama-3.1-8B-Instruct	0.5	14	26/32
	0.6	12	25/32
	0.7	12	25/32
	0.8	9	30/32
Gemma-2-9B-IT	0.5	19	41/42
	0.6	19	41/42
	0.7	19	40/42
	0.8	11	31/42

B.1.2 Outcome adjudication.

Cohort completions are generated deterministically using greedy decoding with at most 128 new tokens and evaluated by the fixed StrongREJECT (Souly et al., 2024) judge. An attacked response y is designated as a successful jailbreak (JS) when

$$s_{\mathrm{SR}}(y) \geq 0.6. \tag{8}$$

We use a fixed operational cutoff, consistent with its use in prior work (Zhu et al., 2026). Responses with intermediate scores are treated as ambiguous and excluded from the primary matched population. A direct harmful request is retained as a DHR anchor only when its completion is independently adjudicated as a refusal. These rules determine cohort membership without using the prompt-boundary $H/R/C$ scores. Probe score and cohort formation are detailed in Appendices B.2 and B.3, respectively.

B.1.3 Sensitivity to the Success Threshold

In the main analysis, we define jailbreak success using $s_{\mathrm{SR}} \geq 0.6$. We consider StrongREJECT cutoffs $\tau \in \{0.5, 0.6, 0.7, 0.8\}$ to assess sensitivity to this choice. For each cutoff, we use the same diagnostic candidate pool and keep model outputs, $H/R/C$ scores, equivalence margins, and statistical procedures fixed.

In Figure 6, we report the layerwise hypothesis diagnosis for the different StrongREJECT cutoffs. For $\tau \in \{0.5, 0.6, 0.7\}$, the joint $H/R/C$ criterion holds in all $28/28$ layers of Qwen2.5-7B-Instruct. For Llama-3.1-8B-Instruct, the corresponding counts are $26/32$, $25/32$, and $25/32$ layers, respectively. Gemma-2-9B-IT meets the joint criterion in $41/42$, $41/42$, and $40/42$ layers over the same cutoffs. The event-level pattern across model depth remains stable near the primary success threshold. At the stricter cutoff $\tau = 0.8$, matched-goal coverage decreases and layerwise support varies more across models.

B.1.4 Cross-adjudication with GPT-4o mini

We use GPT-4o mini as a blinded secondary response judge to assess whether the diagnosis depends on the StrongREJECT decision rule. The judge receives only the original harmful request and the

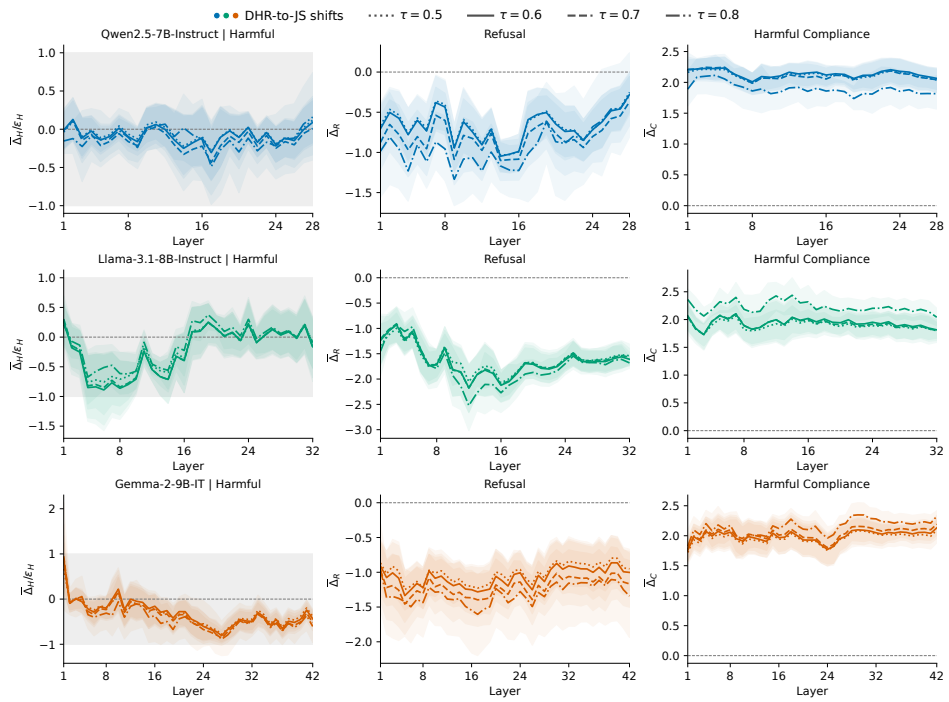

Figure 6: Layerwise sensitivity of the hypothesis diagnosis to the StrongREJECT success threshold across Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, and Gemma-2-9B-IT. The same $H/R/C$ scores and statistical criteria are evaluated under $\tau \in \{0.5, 0.6, 0.7, 0.8\}$. The layerwise redistribution pattern remains stable around the primary cutoff, while the strictest endpoint yields smaller matched populations and more model-dependent support.

saved model response. It classifies each response as substantive harmful assistance, refusal, or neither. Representation scores are not used in adjudication.

GPT-4o mini classifies all $115/115$ primary JS responses as harmful assistance and all $47/47$ DHR anchors as refusals. We then reconstruct the matched populations directly from the secondary labels, without applying the StrongREJECT cutoff. The successful populations increase from $37/16$ to $52/17$ responses/goals for Qwen2.5-7B-Instruct, from $33/12$ to $43/16$ for Llama-3.1-8B-Instruct, and from $45/19$ to $52/19$ for Gemma-2-9B-IT.

We repeat the layerwise hypothesis diagnosis on the reconstructed populations using the same $H/R/C$ scores, equivalence margins, and statistical criteria. Under the secondary adjudication, all three models again meet the predefined full-layer hypothesis criterion. We observe the same pattern: harmfulness is retained, refusal decreases, and compliance increases. Together with the cutoff analysis, these results show that the observed redistribution remains stable under the tested changes to the StrongREJECT cutoff and the response-level rule used to define the successful-jailbreak population.

B.2 Frozen Score Construction and Validation

B.2.1 Operational targets.

Each calibration example consists of a prompt x_i, its completion y_i, and a harmful-goal or benign-request identity. The harmfulness label a_i^H indicates whether the requested objective is harmful. We calibrate the H score using only unwrapped harmful and benign requests. Attack-wrapped prompts are excluded from both fitting and calibration of this score.

The refusal label a_i^R captures explicit refusal, policy invocation, or safety framing. The harmful-compliance label a_i^C indicates substantive unsafe assistance as defined in Appendix B.1. We an-

Table 4: Layerwise hypothesis diagnosis using the primary StrongREJECT endpoint and GPT-4o mini cross-adjudication. Secondary labels change only cohort membership. Layer-specific scores remain frozen, and calibration margins and statistical criteria are unchanged. Entries report the number of layers that meet the pointwise 95% paired goal-bootstrap criteria. Joint denotes the intersection of the passing-layer sets; it does not imply simultaneous confidence coverage.

Model	Adjudication	N_{JS}	G	H	R	C	Joint
Qwen2.5-7B-Instruct	SR-0.6	37	16	28/28	28/28	28/28	28/28
	GPT-4o mini	52	17	28/28	28/28	28/28	28/28
Llama-3.1-8B-Instruct	SR-0.6	33	12	25/32	32/32	32/32	25/32
	GPT-4o mini	43	16	26/32	32/32	32/32	26/32
Gemma-2-9B-IT	SR-0.6	45	19	41/42	42/42	42/42	41/42
	GPT-4o mini	52	19	41/42	42/42	42/42	41/42

notate these two labels independently, allowing refusal cues and harmful assistance to coexist in a response. We exclude responses outside the defined endpoints from R/C fitting.

We fit the R/C scores separately on the same harmful endpoint examples, using completions to provide supervision. All scores take only the pre-generation representation $h(x_i)$ as input.

B.2.2 Goal-disjoint fitting and freezing.

We divide the calibration examples into fit and hold-out sets using an $80/20$ split grouped by harmful-goal or benign-control identity. All realizations of an identity are assigned to the same set. The R/C scores use the same endpoint rows and grouped split, and share an input standardizer fitted on fit examples only. Starting from a fixed seed, we select the first grouped split with both classes represented in both sets. We do not use hold-out performance to select the split.

For each $q \in \{H, R, C\}$, we define

$$\widetilde{h}^{(q)}(x) = \big(h(x) - \mu_q\big) \oslash \sigma_q, \tag{9}$$

$$r_q(x) = \widetilde{w}_q^{\top} \widetilde{h}^{(q)}(x) + b_q, \qquad s_q(x) = \frac{r_q(x) - m_q}{\tau_q}. \tag{10}$$

Here, r_q is the pre-sigmoid output of a class-balanced logistic regression model with ℓ_2 regularization and fixed inverse regularization strength $C = 1$. We standardize its scale using m_q and τ_q. All moments are estimated from fit examples only. We fix the orientation of each score so that larger s_q indicates stronger evidence for target q. The fitted preprocessing and score parameters are frozen before test evaluation. Neither the internal hold-out nor the final DHR–JS goals are used for refitting or selection.

B.2.3 Held-out validity.

In Figure 7, we report layerwise AUROC on identity-grouped calibration hold-outs. For each model, labels and fit/hold-out identity assignments remain fixed across layers. We fit each intermediate-layer probe separately on calibration fit data and freeze it before diagnosis. At the final layer, we evaluate the original frozen score on recaptured hidden states. The R/C probes use the same endpoint rows and share an input scaler within each layer.

For Qwen2.5-7B, Llama-3.1-8B, and Gemma-2-9B, respectively, the H hold-outs contain $37/7$, $37/7$, and $35/7$ examples/identities (N/G). The corresponding shared R/C hold-outs contain $20/4$, $23/4$, and $17/4$.

Across models and layers, held-out AUROC ranges from 0.889 to 1.000 for H, 0.714 to 1.000 for R, and 0.800 to 1.000 for C. These results support held-out discrimination of the operational targets from pre-generation states. Performance varies across layers, particularly for R. The hold-outs contain few identities, so these point estimates should not be interpreted as evidence of perfect generalization or semantic identification. Probe score validity alone does not establish the hypothesis. We test the representational non-redundancy of R/C separately in Sec 2.2.

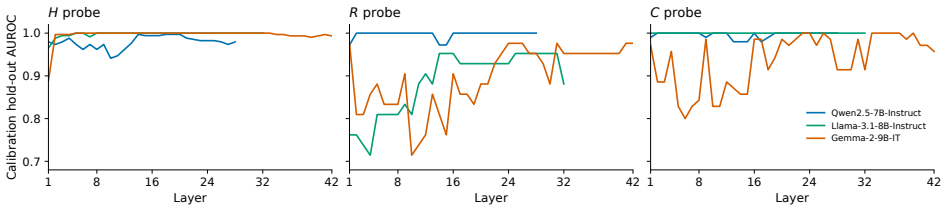

Figure 7: **Layerwise held-out probe score validity.** Curves show AUROC point estimates for the H, R, and C probes on fixed identity-grouped calibration hold-outs. Colors identify models. Layer indices refer to decoder blocks, with the final measurement taken after final normalization. All panels use the same vertical range, $[0.68, 1.02]$. Chance-level AUROC is 0.5, below the displayed range.

B.3 Cohort construction and data roles

B.3.1 Screening and formal adjudication.

We collect released successful attack artifacts from five JailbreakBench attack families: DSN, GCG, JBC, PAIR, and Prompt-with-Random-Search. After removing duplicate prompts, the inventory contains 858 attacks covering 103 harmful goals. We screen this inventory independently on each target model. To limit concentration, we retain at most four screening-time JS prompts per goal and three per goal–attack-family pair. We exclude ambiguous responses and goals with no successful jailbreak.

We pair each retained attack with the direct harmful request for the same goal. We then discard the screening completions and labels and regenerate and re-adjudicate the entire paired cohort under the formal protocol. A DHR instance is retained only if the direct request elicits a strict refusal. An attacked prompt is retained as JS only if its response satisfies the formal harmful-compliance criterion. Ambiguous outcomes are excluded from the final matched cohort.

B.3.2 Partitioning and DHR–JS matching

We assign harmful goals to disjoint discovery, score-calibration, and test partitions using a nominal $50/25/25$ ratio. We fit and freeze scores within the calibration partition as described in Appendix B.2.

For the primary analysis, we include test goals with a retained DHR anchor and at least one formally adjudicated JS response. For Qwen, this gives 53 instances across 16 goals (16 DHR, 37 JS). The corresponding counts are 45 instances across 12 goals for Llama (12 DHR, 33 JS) and 64 instances across 19 goals for Gemma (19 DHR, 45 JS). We count each DHR anchor once. We average JS scores within each goal before computing the DHR-to-JS contrast and use goals as the units of statistical inference (Appendix B.4).

B.4 Goal-level estimation and inference

Model and layer-specific estimands. For each model m, we define $\mathcal{G}_m$ as the set of test goals with a retained DHR anchor and at least one formally adjudicated JS response. For each $g \in \mathcal{G}_m$, let $\mathcal{A}^{\mathrm{JS}}_{g,m}$ be the set of corresponding JS variants and let $N_{g,m} = |\mathcal{A}^{\mathrm{JS}}_{g,m}|$ denote its size. The model has L_m decoder layers, indexed from 1 to L_m. We take the measurement at L_m after the model's final normalization. For $q \in \{H, R, C\}$ and $\ell \in \{1, \ldots, L_m\}$, we define

$$\Delta_{q,m,\ell}(g) = \frac{1}{N_{g,m}} \sum_{a \in \mathcal{A}^{\mathrm{JS}}_{g,m}} \left[s_{q,m,\ell}(x^{\mathrm{JS}}_{g,a}) - s_{q,m,\ell}(x^{\mathrm{DHR}}_{g}) \right], \tag{11}$$

$$\overline{\Delta}_{q,m,\ell} = \frac{1}{|\mathcal{G}_m|} \sum_{g \in \mathcal{G}_m} \Delta_{q,m,\ell}(g). \tag{12}$$

Table 5: Layerwise pointwise criteria. Joint counts use the intersection of all three criteria at each layer. Layer indices are one-based.

Model	Layers	H equiv.	$R\downarrow$	$C\uparrow$	Joint	Joint layers
Qwen2.5-7B	28	28	28	28	28	1–28
Llama-3.1-8B	32	25	32	32	25	1–3, 10–13, 15–32
Gemma-2-9B	42	41	42	42	41	2–42

Table 6: **Refusal–compliance label contingencies.** N_{rc} counts responses annotated as $(R, C) = (r, c)$, and $\nu_{\text{noncomp}} = (N_{00} + N_{11})/N$. Brackets report 95% goal-bootstrap confidence intervals.

Model	Split	N	N_{00}	N_{01}	N_{10}	N_{11}	ν_{noncomp}
Qwen2.5-7B	Cal.	89	2	10	46	31	0.371 [0.282, 0.449]
	Test	83	1	3	43	36	0.446 [0.347, 0.537]
Llama-3.1-8B	Cal.	83	0	18	47	18	0.217 [0.132, 0.300]
	Test	78	1	18	42	17	0.231 [0.132, 0.329]
Gemma-2-9B	Cal.	82	1	10	53	18	0.232 [0.162, 0.293]
	Test	99	0	14	54	31	0.313 [0.236, 0.380]

We use the same goals, DHR anchors, and JS variants at every layer within each model. The first average is over JS variants within a goal. The second is over goals, with each goal receiving equal weight.

Harmfulness-equivalence margins. Let $\mathcal{V}_m^{\text{DHR}}$ and $\mathcal{V}_m^{B}$ denote the direct-harmful and plain-benign anchors from the `validation` partition. Both sets include only records with available responses and `analysis_include=True`. We write $\bar{s}_{H,m,\ell}(\mathcal{V})$ for the record-level mean score and define

$$\epsilon_{H,m,\ell} = \begin{cases} 0.2\left|\bar{s}_{H,m,\ell}(\mathcal{V}_m^{\text{DHR}}) - \bar{s}_{H,m,\ell}(\mathcal{V}_m^{B})\right|, & \ell < L_m, \\ \epsilon_{H,m}^{\text{final}}, & \ell = L_m. \end{cases} \tag{13}$$

At the final measurement point, we retain the previously calibrated margin together with the frozen scoring function and normalization parameters. We use the same final-layer reference in the component and intervention analyses. The final-layer margins are 0.432, 0.427, and 0.434 for Qwen, Llama, and Gemma, respectively. No test records are used to calibrate the margins. We keep scores and margins fixed during bootstrap inference.

Goal-Level Bootstrap Inference and Joint Criterion. We draw 20,000 bootstrap resamples of goals with replacement. Each sampled goal contributes its DHR anchor and all JS variants. For each coordinate, we use the same resampled goals across layers and for both condition means and their paired difference. Let $L_{q,m,\ell}$ and $U_{q,m,\ell}$ denote the 2.5th and 97.5th percentiles of the bootstrap distribution of $\overline{\Delta}_{q,m,\ell}$. These are two-sided, pointwise 95% intervals, not simultaneous confidence bands across layers. We use the paired-difference intervals for inference. The separately plotted DHR and JS condition-mean intervals are not intervals for this difference. The joint hypothesis is satisfied at model-layer pair (m, ℓ) when

$$L_{H,m,\ell} \geq -\epsilon_{H,m,\ell}, \quad U_{H,m,\ell} \leq \epsilon_{H,m,\ell}, \quad U_{R,m,\ell} < 0, \quad L_{C,m,\ell} > 0. \tag{14}$$

We apply these criteria to goal-balanced mean changes. They do not imply that every individual goal satisfies the same conditions.

The joint criterion holds at all evaluated layers in Qwen and at most evaluated layers in Llama and Gemma. These results summarize goal-balanced mean changes across model depth.

B.5 Additional Refusal–Compliance Diagnostics

Table 6 reports the complete calibration and test contingencies underlying Figure 2. The $(1, 1)$ cell accounts for most non-complementary observations, whereas $(0, 0)$ responses are rare.

Using the notation of Appendix B.2, the gradient of the frozen standardized score in raw hidden-state coordinates is

$$\beta_q = \frac{\widetilde{w}_q \oslash \sigma_q}{\tau_q}, \qquad \widehat{\beta}_q = \frac{\beta_q}{\|\beta_q\|_2}, \qquad q \in \{H, R, C\}, \tag{15}$$

where τ_q is the frozen score-standardization scale. To test whether common alignment with H explains the observed R/C geometry, define

$$P_H^{\perp} = I - \widehat{\beta}_H \widehat{\beta}_H^{\top}, \qquad \rho_{RC}^{H^{\perp}} = \frac{(P_H^{\perp}\beta_R)^{\top}(P_H^{\perp}\beta_C)}{\|P_H^{\perp}\beta_R\|_2 \|P_H^{\perp}\beta_C\|_2}. \tag{16}$$

Removing the frozen H direction changes ρ_{RC} by at most 0.005. As a pipeline check, refitting on the same rows with the forced target $a_i^{C^\star} = 1 - a_i^R$ recovers $\rho_{RC^\star} = -1$ to numerical precision for every model. The diagnostic therefore recovers the exact inverse-axis geometry when it is imposed, whereas the observed frozen directions remain far from $|\rho_{RC}| = 1$.

C Implementation and Reproducibility Details

This appendix provides implementation and reproducibility details for Section 3. Unless otherwise stated, the main analysis uses random seed 42; alternative seeds are used only in explicitly reported robustness experiments. All language-model parameters remain frozen throughout. Operational scores, ConCA dictionaries, component estimators, and intervention artifacts are constructed separately for each target model.

C.1 Data Roles and Evaluation Boundaries

Cohort screening, response adjudication, and DHR–JS matching are described in Appendix B.3. Here we define the access boundaries used by the structural and functional experiments and report their model-specific realization.

Structural roles. The discovery, calibration, and test partitions introduced in Appendix B.3 correspond to

$$\mathcal{D}_{\text{disc}} = \mathcal{D}_U \dot{\cup} \mathcal{D}_A, \qquad \mathcal{D}_{\text{cal}} = \mathcal{D}_H, \qquad \mathcal{D}_{\text{test}} = \mathcal{D}_{\text{test}}^{\text{str}}. \tag{17}$$

Thus, D_U and D_A form an internal subdivision of the discovery partition. Harmful records are grouped by harmful-goal identity, whereas benign and style-only records are grouped by benign-control identity. Source-goal and control identities are not jointly constrained across this internal subdivision. This subdivision does not cross the score-calibration or structural-test boundaries.

Table 7: **Data roles and permitted use.** Structural roles are separated by harmful-goal or benign-control identity; functional confirmation is separated from development by normalized prompt identity.

Role	Permitted use	Evaluation status
$\mathcal{D}_U$	Dictionary learning	Structural development
$\mathcal{D}_A$	Component screening	Structural development
$\mathcal{D}_H$	Score fitting, margin calculation, and estimator selection	Structural development
$\mathcal{D}_{\text{test}}^{\text{str}}$	Single–multi fidelity comparison	Structural evaluation
$\mathcal{D}_R$	Restoration and sensitivity construction	Functional development
$\mathcal{D}_{\text{dev}}$	Intervention configuration selection	Functional development
$\mathcal{D}_{\text{test}}^{\text{func}}$	ASR and H-equivalence	Functional confirmation

Let

$$\mathcal{S}_{\text{str}} = \left\{\mathcal{D}_U, \mathcal{D}_A, \mathcal{D}_H, \mathcal{D}_{\text{test}}^{\text{str}}\right\}.$$

Here, the grouping identity is the harmful-goal identity for harmful records and the benign-control identity for B/SO records. For any distinct $\mathcal{D}_i, \mathcal{D}_j \in \mathcal{S}_{\text{str}}$, we enforce

$$\text{Id}(\mathcal{D}_i) \cap \text{Id}(\mathcal{D}_j) = \varnothing. \tag{18}$$

All realizations sharing the corresponding grouping identity remain together. Because score calibration and component-estimator selection both access $\mathcal{D}_H$, they form one development stage instead of independent evidence. Attack families may recur across partitions, so structural evaluation measures transfer to unseen identities within the screened population, not to unseen attack families.

Functional boundary. The functional branch uses $\mathcal{D}_R$ for restoration and sensitivity construction and $\mathcal{D}_{\text{dev}}$ for configuration selection. Its final confirmation cohort satisfies

$$P_{\text{norm}}\big(\mathcal{D}^{\text{func}}_{\text{test}}\big) \cap P_{\text{norm}}(\mathcal{D}_R \cup \mathcal{D}_{\text{dev}}) = \varnothing. \tag{19}$$

Harmful goals and attack families may overlap across partitions, so functional confirmation tests transfer to unseen prompts, not unseen goals or attack families. Membership is fixed before baseline generation and does not depend on baseline success. All retained prompts are included in the ASR denominator.

The structural test is identity-disjoint from estimator development but remains conditional on the target-model-screened, outcome-enriched population. The functional test is prompt-disjoint from restoration development but does not impose unseen-goal or unseen-family separation.

Table 8: **Model-specific evaluation cohorts.** Structural columns report prompt records and harmful-goal or benign-control identities (N/E); the functional column reports attacked prompts and harmful-goal clusters (N/G). Structural representation counts may include records without strict R/C endpoints.

Model	$\mathcal{D}_U$ (N/E)	$\mathcal{D}_A$ (N/E)	$\mathcal{D}_H$ (N/E)	$\mathcal{D}^{\text{str}}_{\text{test}}$ (N/E)	$\mathcal{D}^{\text{func}}_{\text{test}}$ (N/G)
Qwen2.5-7B	273/54	59/13	163/34	161/33	532/99
Llama-3.1-8B	243/50	60/13	157/32	150/31	564/103
Gemma-2-9B	278/55	67/14	157/34	174/34	517/98

C.2 Representation Extraction and Generation

Prompt serialization. Each input is represented as a single user turn and serialized using the target model's native chat template with the assistant-generation prefix appended. Padding is applied only for batching and does not alter the location of the analyzed state.

Prompt-boundary state. For an input x, we perform a prompt-only forward pass and extract the hidden state at the last non-padding prompt token. For the structural analysis, this is the final-layer state

$$h(x) = h_{L,T}(x),$$

as defined in Section 3. Intermediate-layer states used for functional localization are extracted at the same prompt boundary. All representations are captured before the first completion token is generated.

Extracted hidden states and downstream ConCA coordinates are stored and processed in FP32. No completion-side state is used to train the operational score or the structural component estimators.

Generation settings. All cohort-construction and evaluation responses are generated deterministically using greedy decoding. Responses used in the structural cohort contain at most 128 newly generated tokens. Independent functional-confirmation runs allow at most 256 newly generated tokens and use an unquantized BF16 model with maximum context length 2,048. Within each experiment, the baseline, intervention, and control conditions use identical tokenization, context handling, stopping rules, and decoding parameters.

Screening continuations used during cohort enrichment are discarded. After the model-specific cohort has been fixed, responses are regenerated under the formal evaluation configuration.

C.3 ConCA Training

We instantiate one LayerNorm ConCA decomposition per target model using only prompt-boundary representations in $\mathcal{D}_U$. Each dictionary contains $d_z = 4096$ components. Let $z_\theta(h)$ denote the dense

Configuration	Value
Components d_z	4096
Updates	5000
Batch size	128
Sampling	With replacement
Learning rate	2×10^{-4}
LR warm-up	500
Final λ_t	0.02
Sparsity warm-up	2000
Regularized coordinates	$\mathrm{SELU}(z_\theta(h))$
Downstream coordinates	$z_\theta(h)$

Table 9: Shared ConCA training configuration.

encoder coordinates and $\widehat{h}_\theta(h)$ the reconstructed representation. We optimize

$$\mathcal{L}_{\mathrm{ConCA}} = \frac{1}{B}\sum_{i=1}^{B}\left\|h_i - \widehat{h}_\theta(h_i)\right\|_2^2 + \lambda_t \frac{1}{B}\sum_{i=1}^{B}\left\|\mathrm{SELU}(z_\theta(h_i))\right\|_1 . \tag{20}$$

SELU is used only to define the sparsity penalty. Component screening, harmfulness alignment, and all downstream analyses use the dense coordinates $z_\theta(h)$; no top-k, thresholding, or post-hoc sparsification is applied.

The shared training configuration is reported in Table 9.

C.4 Component Screening and Estimation

We construct the H-aligned component estimator in three stages: candidate screening on $\mathcal{D}_A$, entity-grouped selection on $\mathcal{D}_H$, and held-out evaluation on $\mathcal{D}_{\mathrm{test}}^{\mathrm{str}}$.

Candidate screening. For component j, let $z_j(x)$ denote its dense coordinate and d_j its decoder atom. We rank components using both their association with the frozen harmfulness score and their decoder alignment with the raw-coordinate score direction β_H:

$$r_j = \left|\rho_S(z_j(x), s_H(x))\right|, \qquad q_j = \frac{|d_j^\top \beta_H|}{\|d_j\|_2 \|\beta_H\|_2}. \tag{21}$$

The candidate set $\mathcal{C}_H$ is the union of the top 64 active coordinates under r_j and q_j, computed once on $\mathcal{D}_A$.

Grouped selection and stability. On $\mathcal{D}_H$, we fit Elastic-Net regressions to the continuous frozen score s_H using up to five folds grouped by harmful-goal or benign-control identity. Standardization is estimated within each training fold. We select (λ, ρ) by grouped held-out R^2 using the one-standard-error rule over 10 logarithmically spaced $\lambda \in [10^{-3}, 1]$ and $\rho \in \{0.5, 0.8, 1.0\}$.

After fixing the selected configuration, we repeat the fit over 100 resamples containing 75% of the calibration entities. For each candidate j, let π_j denote its selection frequency and κ_j the fraction of selected fits agreeing with its modal coefficient sign. The stable support is

$$\mathcal{S}_H = \{j \in \mathcal{C}_H : \pi_j \geq 0.70,\ \kappa_j \geq 0.80\}. \tag{22}$$

Support stability is evaluated conditional on the frozen ConCA dictionary.

Final estimators. After support selection, we refit a Ridge estimator ($\lambda_R = 1$) on all records in $\mathcal{D}_H$ to obtain

$$\widehat{s}_H^{\mathrm{multi}}(x) = \widehat{b} + \sum_{j \in \mathcal{S}_H} \widehat{\gamma}_j z_j(x).$$

The strongest-single comparator is selected separately by fitting the same one-coordinate Ridge estimator to every active ConCA coordinate and choosing the coordinate with the highest entity-grouped

Table 10: Component-H alignment and support selection.

Model	λ_{EN}	ρ	$\lvert\mathcal{S}_H\rvert$
Qwen2.5-7B	0.0464	0.8	21
Llama-3.1-8B	0.0464	1.0	11
Gemma-2-9B	0.0215	1.0	22

development R^2. Thus, the single-component baseline is not restricted by the multi-component candidate screen.

All screening, selection, and estimator fitting are completed before evaluation on $\mathcal{D}^{\mathrm{str}}_{\mathrm{test}}$.

C.5 Component-Protected Natural Restoration

Goal-balanced natural actuator. For a zero-based transformer block ℓ, let $h_{\ell,T}(x)$ denote the post-block residual state (`resid_post`) at the final prompt position. For each development goal g, we pair an available DHR prompt x_g^{DHR} with its n_g strictly successful jailbreak realizations. We define

$$v^{(\ell)}_{\mathrm{nat}} = \frac{1}{|\mathcal{G}_R|} \sum_{g \in \mathcal{G}_R} \left[h_{\ell,T}(x_g^{\mathrm{dir}}) - \frac{1}{n_g} \sum_{a=1}^{n_g} h_{\ell,T}(x_{g,a}^{\mathrm{JS}}) \right], \tag{23}$$

where $\mathcal{G}_R$ contains goals with an available DHR prompt and at least one strictly successful JS realization. We average within each goal so that goals contribute equally. The restoration set $\mathcal{D}_R$ is a matched subset of the development pools. All eligible DHR anchors are included, with no additional response-outcome filtering.

Layer localization. We evaluate eight candidate blocks per model by replacing the JS prompt-boundary state with its same-goal DHR state on development records. The selected zero-based blocks are

$$\ell_{\mathrm{Qwen}} = 20, \qquad \ell_{\mathrm{Llama}} = 19, \qquad \ell_{\mathrm{Gemma}} = 36.$$

For the scalar score comparator, we stack the pullback sensitivities of the frozen harmfulness score,

$$G^{(\ell)}_{\mathrm{lin}} = \mathrm{rowstack}_{x_i \in \mathcal{D}_R} \left[\nabla_{h_{\ell,T}(x_i)} s_H(x_i) \right]^\top = U_{\mathrm{lin}} \Sigma_{\mathrm{lin}} V_{\mathrm{lin}}^\top, \tag{24}$$

and let

$$Q^{(\ell)}_{\mathrm{lin}} = V_{\mathrm{lin}}[:, 1 : r_{\mathrm{lin}}] \tag{25}$$

denote the retained scalar score sensitivity subspace.

Alignment-weighted component pullbacks. Let $\mathcal{S}_H$ denote the stable set of H-aligned components selected by the preceding structural procedure, and let

$$z_j(x) = z_{\theta,j}(h_{L,T}(x))$$

denote the corresponding dense final-layer component coordinate. We use the magnitude of the raw-coordinate slope b_j to weight the component-wise pullbacks:

$$b_j = \sigma_{s_H} \frac{\widehat{\gamma} * j}{\sigma * z, j}, \qquad \omega_j = \left(\frac{|b_j|}{|\mathcal{S}_H|^{-1} \sum_{k \in \mathcal{S}_H} |b_k|} \right)^{1/2}, \qquad \phi_j(x) = \omega_j z_j(x). \tag{26}$$

All means and scales are frozen from the calibration fit set. ϕ_j is therefore a square-root alignment-weighted component quantity, with components differentiated separately before scalar reconstruction.

For every restoration record and selected component, we pull the component quantity back to the intervention block:

$$G^{(\ell)}_{\mathrm{SC}} = \mathrm{rowstack}_{\substack{x_i \in \mathcal{D}_R \\ j \in \mathcal{S}_H}} \left[\omega_j \nabla_{h_{\ell,T}(x_i)} z_j(x_i) \right]^\top. \tag{27}$$

Both DHR and strictly successful JS records enter this matrix. Each prompt–component pair contributes one row; no cross-record averaging, centering, row normalization, or whitening is applied. Let

$$G_{\mathrm{SC}}^{(\ell)} = U_{\mathrm{SC}}\Sigma_{\mathrm{SC}}V_{\mathrm{SC}}^{\top}, \qquad Q_{\mathrm{SC}}^{(\ell)} = V_{\mathrm{SC}}[:, 1 : r_H]. \tag{28}$$

The applied truncation ranks are $r_H = 9, 4, 2$ for Qwen, Llama, and Gemma, respectively, and differ from the native 90%-energy ranks.

Protected restoration operator. The score-gated comparison operator uses the unprojected direction

$$v_{\mathrm{unproj}}^{(\ell)} = v_{\mathrm{nat}}^{(\ell)}. \tag{29}$$

The Linear-H-protected operator removes the scalar-score pullback subspace and restores the original intervention norm:

$$\widetilde{v}_{\mathrm{lin}}^{(\ell)} = \left(I - Q_{\mathrm{lin}}^{(\ell)}Q_{\mathrm{lin}}^{(\ell)\top}\right) v_{\mathrm{nat}}^{(\ell)}, \tag{30}$$

$$v_{\mathrm{lin}}^{(\ell)} = \frac{\|v_{\mathrm{nat}}^{(\ell)}\|_2}{\|\widetilde{v}_{\mathrm{lin}}^{(\ell)}\|_2}\widetilde{v}_{\mathrm{lin}}^{(\ell)}. \tag{31}$$

The component-protected operator removes the component-pullback subspace and restores the original intervention norm:

$$\widetilde{v}_{\mathrm{SC}}^{(\ell)} = \left(I - Q_{\mathrm{SC}}^{(\ell)}Q_{\mathrm{SC}}^{(\ell)\top}\right) v_{\mathrm{nat}}^{(\ell)}, \tag{32}$$

$$v_{\mathrm{SC}}^{(\ell)} = \frac{\|v_{\mathrm{nat}}^{(\ell)}\|_2}{\|\widetilde{v}_{\mathrm{SC}}^{(\ell)}\|_2}\widetilde{v}_{\mathrm{SC}}^{(\ell)}. \tag{33}$$

Score gate and application. Let $s_H(x)$, $s_R(x)$, and $s_C(x)$ denote the frozen standardized scores computed from the unmodified final-layer prompt-boundary state. For model m, the deployment gate is

$$\chi_m(x) = \mathbf{1}\left[s_H(x) > \tau_{H,m} \ \wedge \ s_C(x) - s_R(x) > \tau_{RC,m}\right]. \tag{34}$$

The gate is computed once before generation and shared by the two natural-restoration operators within each model.

For the natural-restoration experiments, the intervention is applied once during prompt prefill to the final prompt-token block output:

$$h_{\ell,T} \leftarrow h_{\ell,T} + \chi_m(x)\,\alpha_{m,o}\,v_o^{(\ell)}, \qquad o \in \{\mathrm{unproj}, \mathrm{lin}, \mathrm{SC}\}. \tag{35}$$

We fix $\alpha_{m,o} = 1$ for all models and natural-restoration operators. Thus, the main natural-restoration setting is prefill-only: no intervention is applied to subsequently generated tokens. If the gate is inactive, no state is modified and generation follows the baseline. Gate activation does not change membership in the ASR denominator.

Relation to the structural branch. The goal-matched DHR–JS displacement defines the restoration actuator. Component-H does not construct this direction; its selected, alignment-weighted component pullbacks define the protection subspace removed from the final operator. The R/C scores likewise do not construct the actuator or protection basis, but their difference directly enters the frozen deployment gate.

D Additional Structural Details

D.1 Recovery protocol.

For each model, component-H is first learned from unlabeled activation structure. The frozen operational H score is then used on separate development partitions to align the recovered components

Model	(τ_H, τ_{RC})	ℓ	$\lvert\mathcal{S}_H\rvert/r_H$
Qwen2.5-7B	$(0.0638554, -1.3823748)$	20	21/9
Llama-3.1-8B	$(-0.3156117, -0.7722640)$	19	11/4
Gemma-2-9B	$(-0.0069780, 0.1128409)$	36	22/2

Table 11: Frozen gates and application configurations. Both operators share the same gate within each model. Block indices are zero-based, and all interventions act on the active-token `resid_post` state. $|\mathcal{S}_H|$ denotes the selected component support size, whereas r_H denotes the applied pullback-subspace rank.

Table 12: Held-out ConCA reconstruction fidelity. R^2_{dec} evaluates the frozen harmfulness score on reconstructed states against its original scores. Intervals are 95% paired goal/control-entity cluster bootstrap confidence intervals.

Model	MSE ↓	Cosine ↑	R^2_{dec} [95% CI]
Qwen2.5-7B	1.3999	0.9697	0.980 [0.959, 0.993]
Llama-3.1-8B	0.7215	0.9290	0.983 [0.964, 0.994]
Gemma-2-9B	0.4435	0.9498	0.970 [0.952, 0.983]

and select the single- and multi-component estimators. We use the same fixed absolute-L_1 configuration across models.

The data are divided into recovery, alignment, calibration, and final test partitions, with no entity overlap between partitions. Support selection uses 100 entity-level resamples and a fixed selection frequency threshold of 0.7. The best single component and the multi-component support are selected from the same frozen dictionary under the same entity-grouped development criterion. All choices are frozen before final test evaluation.

D.2 Reconstruction fidelity.

We evaluate the frozen ConCA reconstruction on the same structural test entities used for component recovery. We report per-coordinate reconstruction MSE, mean cosine similarity, and the fidelity of the decoder-induced harmfulness score s_H^{dec}:

$$R^2_{\text{dec}} = 1 - \frac{\sum_i [s_H(x_i) - s_H^{\text{dec}}(x_i)]^2}{\sum_i [s_H(x_i) - \overline{s_H}]^2}. \tag{36}$$

The decoder, score, and all preprocessing remain fixed. Confidence intervals use paired resampling of harmful-goal and benign-control identities. These estimates assess reconstruction fidelity separately from the supervised single- and multi-component score estimators.

D.3 Dictionary-seed robustness

We retrain ConCA with three random seeds (41, 42, and 43), holding data partitions, the frozen H scoring function, and dictionary architecture fixed. Component selection and fitting of both single- and multi-component estimators are repeated for each dictionary using the same protocol.

The multi-component estimator achieves higher held-out R^2 in all nine model–seed runs (Figure 8), with all paired entity-cluster bootstrap 95% confidence intervals for ΔR^2 lying above zero.

For the selected decoder matrices with unit-normalized columns, $\sigma_{\min}/\sigma_1 > 0.1$ in every run (Figure 9), supporting non-rank-one geometry across the tested initializations.

Seed 42 provides the reference dictionary for the main structural analysis and intervention experiments.

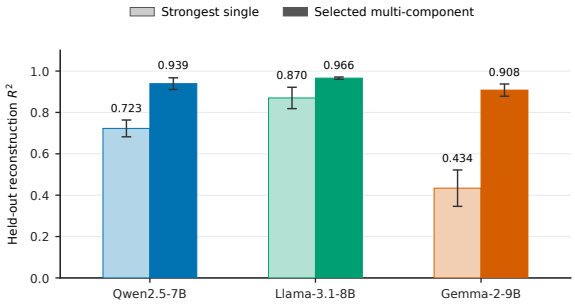

Figure 8: Reconstruction of the frozen H score across dictionary seeds. Bars show mean held-out R^2 for the strongest-single (light) and multi-component (dark) estimators over seeds 41, 42, and 43. Error bars indicate one sample standard deviation.

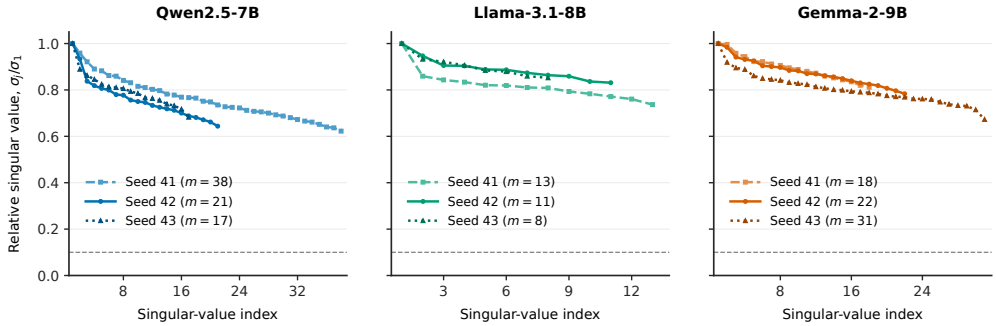

Figure 9: Selected decoder spectra across dictionary seeds. Each panel compares seeds 41, 42, and 43, showing relative singular values σ_j/σ_1 after normalizing decoder columns to unit length. The dashed line marks 0.1; m denotes the selected support size.

D.4 Held-out recovery.

Table 13 reports the complete held-out results. Confidence intervals for ΔR^2 are obtained with a paired entity-cluster bootstrap, evaluating the frozen single- and multi-component estimators on the same resampled entities. R^2 is the primary fidelity measure; the remaining metrics provide complementary checks of score recovery and ranking.

All supplementary metrics follow the primary R^2 result: multi-component recovery improves score fidelity and rank agreement for every model.

D.5 Decoder geometry.

For a selected support S, we form the unit-normalized decoder matrix

$$\widetilde{\mathbf{D}}_S = \left[\frac{\mathbf{d}_j}{\|\mathbf{d}_j\|_2}\right]_{j\in S}, \tag{37}$$

where $\mathbf{d}_j$ is the decoder direction of component j. Figure 4 reports the corresponding singular-value spectra. Table 14 provides scalar summaries of the same geometry.

Across models, the selected decoder matrices retain multiple substantial spectral modes and exhibit low pairwise coherence.These diagnostics establish non-rank-one structure only within the learned Component-H decomposition and under the chosen dictionary, logit score, and selection procedure; they do not identify the intrinsic dimensionality of harmfulness.

The natural displacement is constructed independently of component selection, while the selected component sensitivities determine the protection constraint.

Table 13: Held-out recovery of the frozen operational H score. Arrows denote best single-component to multi-component performance.

Model	R^2	Spearman	sRMSE	AUROC	ΔR^2 [95% CI]
Qwen2.5-7B	$0.702 \rightarrow 0.935$	$0.822 \rightarrow 0.944$	$0.546 \rightarrow 0.256$	$0.937 \rightarrow 0.992$	$+0.232\,[0.146,\ 0.372]$
Llama-3.1-8B	$0.912 \rightarrow 0.964$	$0.892 \rightarrow 0.930$	$0.296 \rightarrow 0.189$	$0.998 \rightarrow 0.999$	$+0.052\,[0.028,\ 0.086]$
Gemma-2-9B	$0.535 \rightarrow 0.939$	$0.598 \rightarrow 0.883$	$0.682 \rightarrow 0.248$	$0.828 \rightarrow 0.991$	$+0.403\,[0.274,\ 0.609]$

Table 14: Geometry of the unit-normalized selected decoder directions. Stable rank is $\|\widetilde{\mathbf{D}}_S\|_F^2/\|\widetilde{\mathbf{D}}_S\|_2^2$; participation rank is computed from the normalized squared singular values.

Model	Selected m	Numerical rank	Stable rank	Participation rank	Max $\lvert\cos\rvert$
Qwen2.5-7B	21	21	12.256	19.872	0.185
Llama-3.1-8B	11	11	8.746	10.875	0.115
Gemma-2-9B	22	22	16.925	21.637	0.111

E Additional Functional Evaluation

E.1 Functional Evaluation Protocol

Functional evaluation uses exact-prompt-unseen attack prompts relative to restoration construction and configuration selection. Harmful goals and attack families may recur. Cohort membership is fixed independently of baseline jailbreak success, and all retained prompts enter the full-cohort ASR denominator. Baseline and intervention conditions use identical generation and adjudication procedures.

The primary behavioral endpoint is full-cohort prompt-level ASR. We additionally report gate-active ASR and frozen $H/R/C$ changes on prompts that actually receive the intervention. Gate activity is computed once from the unmodified prompt-boundary state and held fixed across paired baseline and intervention conditions.

Full-cohort and gate-active estimates. For evaluation prompt x_i, let $a_i = \chi(x_i)$ denote the frozen gate decision and $\mathcal{A} = \{i : a_i = 1\}$. For $q \in \{H, R, C\}$, define $\delta_{q,i} = s_q^{\text{int}}(x_i) - s_q^{\text{base}}(x_i)$. We report

$$\overline{\Delta}_{q,\text{full}} = \frac{1}{N}\sum_{i=1}^{N}\delta_{q,i}, \qquad \overline{\Delta}_{q,\text{active}} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}\delta_{q,i}. \tag{38}$$

Full-cohort ASR uses all evaluation prompts, whereas active-set ASR uses the same fixed set $\mathcal{A}$ for baseline and intervention conditions.

Confidence intervals use 20,000 paired goal-cluster bootstrap replicates. Each sampled goal retains all associated attack variants and their fixed gate decisions. Point estimates weight prompts equally; for gate-active analyses, the active denominator is recomputed within each bootstrap replicate.

For each model, the harmfulness-equivalence margin $\epsilon_{H,m}$ is computed from the calibration anchors defined in Appendix B.4 and reused unchanged throughout functional evaluation. We report H-equivalence for both the full cohort and the gate-active subset, requiring the corresponding 95% confidence interval for the mean intervention-induced H shift to remain within $[-\epsilon_{H,m}, \epsilon_{H,m}]$.

Response-level audits distinguish the full baseline-successful cohort from the gate-active baseline-successful subset; reported rates use the corresponding set as their denominator. Frozen operator and gate configurations are given in Appendix C.5.

Response-audit denominators. Let $\mathcal{S}_0$ denote the prompts that satisfy the strict-success criterion at baseline and $\mathcal{S}_{\text{active}} = \mathcal{S}_0 \cap \mathcal{A}$. Full-cohort response audits use $\mathcal{S}_0$, while active-set safe-reversal and degeneration rates use $\mathcal{S}_{\text{active}}$.

E.1.1 SEMANTIC VALIDATION OF STRICT-SUCCESS REVERSALS.

Semantic judgments are matched to the strict-success outcomes of the same generated responses. For operator o, let $b_i^{(o)}$ indicate strict success after intervention, and let S_0 denote the baseline-successful prompts. The set of strict-success reversals is

$$\mathcal{V}_o = \left\{ i \in S_0 : b_i^{(o)} = 0 \right\}. \tag{39}$$

Let $\mathrm{Safe}(y) = 1$ when the response is coherent, refuses the harmful goal or provides a safe alternative, and contains no substantive harmful assistance. Responses that combine refusal or safety framing with harmful assistance are labeled `MIXED_PARTIAL` and do not satisfy this criterion. We report the fraction of reversals with coherent safe responses:

$$\widehat{p}^{(o)}_{\mathrm{safe|rev}} = \frac{\sum_{i \in \mathcal{V}_o} \mathrm{Safe}\left(y_i^{(o)}\right)}{|\mathcal{V}_o|}. \tag{40}$$

Numerators and denominators are reported alongside the percentages. Full-cohort post-intervention ASR continues to use all evaluation prompts as its denominator.

E.1.2 WHY PRESERVE HARMFULNESS RECOGNITION?

H-preservation follows from the event-level characterization. In our analysis, successful jailbreaks occur with harmfulness recognition approximately unchanged, while refusal decreases and harmful compliance increases. The intervention therefore targets the response-policy factors R/C while keeping H stable.

The H constraint also isolates the effect of the intervention. Without it, a reduction in ASR could result from changing harmfulness representations not reversing the identified response-policy shift.

The current refusal-direction control does not establish that changes in H cause utility loss. For Qwen and Gemma, large changes in H coincide with benign over-refusal and response-quality degradation, but that control also differs in direction, gating, and application schedule. We therefore treat H-preservation as a specificity constraint derived from the event-level hypothesis, not as evidence that changing H is itself harmful.

E.2 FULL-COHORT AND GATE-ACTIVE NATURAL RESTORATION

Table 16 reports full-cohort jailbreak suppression, and the corresponding operational H-equivalence results. All three variants use the same frozen deployment gate. The unprojected operator isolates the effect of the empirical natural actuator without a directional H-constraint; Linear-H protection provides a scalar score baseline; and Component-H protection tests the component-resolved constraint.

Table 15 isolates the prompts that actually receive the intervention. Both operators use the same frozen gate within each model, with active membership determined from the unmodified prompt-boundary state. Both methods continue to satisfy H-equivalence on the active subsets, while R increases and C decreases across all three models. The R/C changes serve as manipulation checks of the intended response-policy restoration.

Table 18 reports the semantic validation of natural-restoration reversals alongside targeted steering, using the common criterion defined above.

The diagnostic cohorts contain baseline-successful, gate-active prompts with goals disjoint from score calibration: 111/63, 100/56, and 136/63 prompts/goals for Qwen, Llama, and Gemma. Intermediate scores are calibrated on separate calibration data, while the final layer uses the primary frozen score. Estimates average prompt variants within goals, weight goals equally, and use paired goal-bootstrap confidence intervals.

Figure 10 shows higher refusal and lower harmful-compliance scores after the intervention, with both shifts persisting to the final layer across all three models. The final $\Delta H/\epsilon_H$ values are $-0.020\,[-0.074,\ 0.034]$ for Qwen, $-0.265\,[-0.326,\ -0.211]$ for Llama, and $0.643\,[0.612,\ 0.673]$ for Gemma. All intervals lie within the prespecified equivalence region $[-1, 1]$.

Table 15: Gate-active natural-restoration results. Baseline and intervention quantities are evaluated on the same prompts selected by the frozen pre-intervention gate. All H-equivalence intervals lie within the prespecified region $[-1, 1]$.

Model	Operator	Active (N/G)	Base ASR (%)	Post-ASR (%)	$\overline{\Delta H}_{\text{active}}/\epsilon_H$ [95% CI]
Qwen2.5-7B	Unprojected natural	348/94	39.94	16.38	0.278 [0.246, 0.310]
	Linear-H-protected			18.97	−0.071 [−0.101, −0.040]
	Component-H-protected			**15.23**	−0.015 [−0.052, 0.021]
Llama-3.1-8B	Unprojected natural	291/97	40.21	3.78	0.012 [−0.059, 0.080]
	Linear-H-protected			2.75	0.206 [0.156, 0.255]
	Component-H-protected			**2.06**	−0.500 [−0.556, −0.447]
Gemma-2-9B	Unprojected natural	271/90	62.36	55.35	0.729 [0.692, 0.764]
	Linear-H-protected			55.35	−0.315 [−0.345, −0.286]
	Component-H-protected			**53.14**	0.617 [0.581, 0.651]

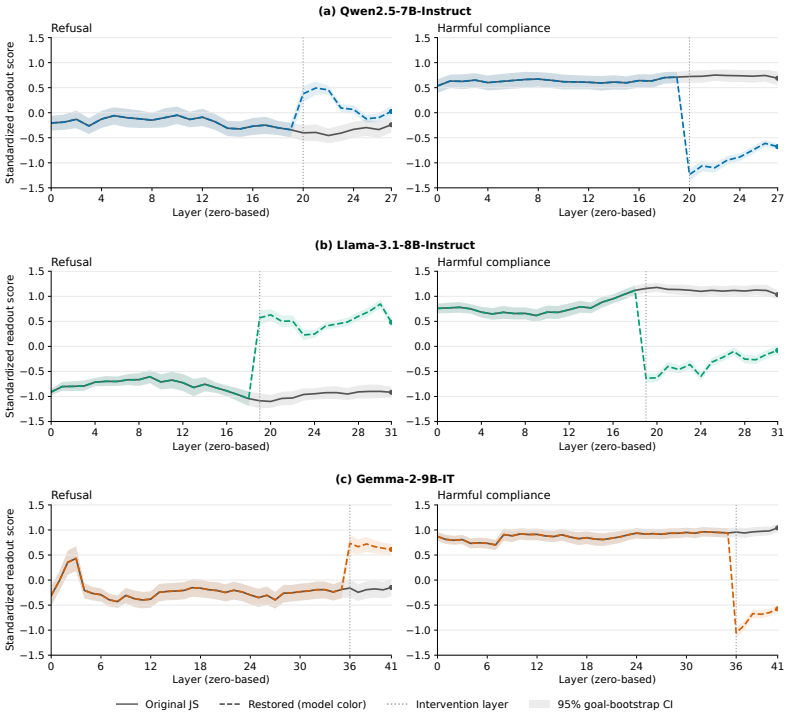

Figure 10: Layerwise response-policy changes under frozen Component-H-protected natural restoration. Rows show Qwen, Llama, and Gemma; columns show refusal and harmful compliance. Solid gray and colored dashed curves show original and restored prompt-boundary score during the initial forward pass. Vertical dotted lines mark intervention layers. Shading shows pointwise 95% goal-bootstrap confidence intervals.

E.3 Direction Controls and Semantic Safety

Projected-random direction controls. For each model, we sample 20 random directions, apply the same H-protection projection as natural restoration, and match their intervention energy. Evaluation uses a separate cohort containing only originally successful jailbreaks, with baseline ASR of 100%. Natural restoration yields lower ASR than every random draw for all three models (Figure 11).

Semantic safety of reversals. Using an independent automated semantic audit, we evaluate responses that change from strict success to non-success under Component-H natural restoration on the full attack cohorts. The proportion judged coherent and free of substantive harmful assistance is 85.85% for Qwen, 95.54% for Llama, and 36.59% for Gemma. Among Gemma's 41 reversals, 15

Table 16: Full-cohort natural-restoration results. ASR is reported in percent. $\overline{\Delta H}_{\text{full}}/\epsilon_H$ is the normalized mean harmfulness shift with 95% paired goal-bootstrap confidence intervals. H-equivalence requires the complete interval to lie within $[-1, 1]$. Bold indicates the lowest post-intervention ASR within each model.

Model	Operator	Baseline ASR (%)	Post-ASR (%)	$\overline{\Delta H}_{\text{full}}/\epsilon_H$ [95% CI]
Qwen2.5-7B	Unprojected natural	35.53	20.11	0.182 [0.158, 0.206]
	Linear-H-protected		21.80	−0.046 [−0.066, −0.026]
	Component-H-protected		**19.36**	−0.010 [−0.033, 0.014]
Llama-3.1-8B	Unprojected natural	22.16	3.37	0.006 [−0.031, 0.041]
	Linear-H-protected		2.84	0.106 [0.080, 0.132]
	Component-H-protected		**2.48**	−0.258 [−0.293, −0.225]
Gemma-2-9B	Unprojected natural	35.78	32.11	0.382 [0.353, 0.410]
	Linear-H-protected		32.11	−0.165 [−0.184, −0.147]
	Component-H-protected		**30.95**	0.323 [0.297, 0.349]

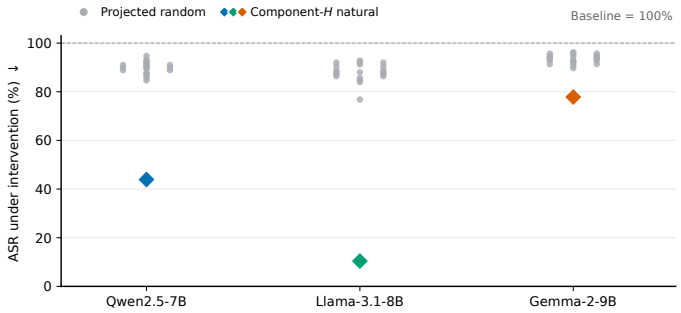

Figure 11: Post-intervention ASR for natural restoration and projected-random controls. Gray points represent 20 random directions; model-colored diamonds denote natural restoration.

are coherently safe, 24 retain harmful assistance, and two exhibit generation degeneration. Its low safe-reversal fraction therefore primarily reflects residual harmful assistance.

E.4 Targeted Steering under Multi-Component H Protection

The main-text experiment fixes the empirical $DHR - JS$ displacement to test natural state restoration. To examine whether the same Component H-constraint can support a purpose-built defensive actuator, we replace the natural displacement with a targeted response-policy direction.

E.4.1 Component-based H protection and application.

We apply the Component harmfulness constraint to the selected actuator using

$$\widetilde{v}_{\text{tar}} = \left(I - \gamma Q_{H,\text{tar}} Q_{H,\text{tar}}^{\top}\right) a_{\text{tar}}, \qquad v_{\text{tar}} = \frac{\|a_{\text{tar}}\|_2}{\|\widetilde{v}_{\text{tar}}\|_2} \widetilde{v}_{\text{tar}}, \tag{41}$$

where $\gamma \in [0, 1]$ controls the strength of harmfulness protection. We use $\gamma = 1$ for Qwen and Gemma and $\gamma = 0.25$ for Llama.

Because targeted steering acts at the transformer-block input and over all active token positions, whereas natural restoration acts at the block output on the active last token, its protection basis $Q_{H,\text{tar}}$ is recomputed under the corresponding application schedule. Specifically, the component pullbacks in Eq. 27 are aggregated over the token positions receiving the targeted intervention before the truncated sensitivity subspace is formed. Thus, natural restoration and targeted steering share the same component-based harmfulness-protection principle, but differ in both actuator construction and application schedule.

Table 17 compares Component natural restoration with targeted steering. ASR is evaluated on the same full attack cohorts used for natural restoration, whereas steering-induced H changes are eval-

Table 17: Targeted steering under the Component-H constraint. ASR is evaluated on the full frozen attack cohorts, while H-preservation is evaluated on gate-active prompts.

Model	Natural ASR (%)	Steering ASR (%)	$\overline{\Delta H}^{\text{steer}}_{\text{active}}/\epsilon_H$ [95% CI]	Degeneration (%)	Utility
Qwen2.5-7B	19.36	17.29	$-0.872\,[-0.935,\ -0.804]$	8.47	Pass
Llama-3.1-8B	2.48	1.42	$0.499\,[0.409,\ 0.588]$	0.00	Pass
Gemma-2-9B	30.95	10.64	$0.821\,[0.782,\ 0.856]$	1.08	Pass

Table 18: Semantic validation of strict-success reversals. Both operators use Component H-protection. Post-ASR is evaluated on the full attack cohort. Safe/reversed counts coherent safe responses among baseline-successful attacks that no longer satisfy strict success after the indicated intervention. The final column reports this conditional fraction.

Model	Operator	Full-cohort post-ASR (%)	Safe / reversed	Safe fraction (%)
Qwen	Natural restoration	19.36	91/106	85.85
	Targeted steering	17.29	104/117	88.89
Llama	Natural restoration	2.48	107/112	95.54
	Targeted steering	1.42	117/117	100.00
Gemma	Natural restoration	30.95	15/41	36.59
	Targeted steering	10.64	133/139	95.68

uated on the corresponding gate-active prompts. All three active-set H intervals remain within the fixed equivalence region.

Targeted steering further reduces post-intervention ASR from 19.36% to 17.29% for Qwen, from 2.48% to 1.42% for Llama, and from 30.95% to 10.64% for Gemma. Degeneration, evaluated over the baseline-successful cohort, is 8.47%, 0.00%, and 1.08%, respectively. All three gated steering configurations satisfy the pooled benign/style-only utility criterion; complete utility results are reported in Appendix E.5.

Table 18 compares the semantic outcomes of strict-success reversals under natural restoration and targeted steering. Coherent safe responses account for most targeted-steering reversals in all three models. By contrast, under natural restoration, Gemma's low safe-reversal fraction primarily reflects residual harmful assistance. Among the 41 natural-restoration reversals, the automated semantic audit identifies 24 responses that retain harmful assistance and only two that exhibit degeneration. Of those 24, 22 are classified as partial under the strict evaluation: they leave the strict-success category without becoming semantically safe. Natural restoration therefore reduces strict ASR but yields coherent safe responses in only 15/41 reversals (36.59%).

E.5 Benign and Style-Only Utility

The utility set contains 60 held-out benign requests from WildJailbreak (Jiang et al., 2024). Each request is evaluated in plain form and under one wrapper from each of five attack families, yielding 60 plain and 300 style-only prompts per model. All wrapped variants preserve the benign objective. Intervention settings remain frozen, with no utility-side tuning.

We evaluate paired responses under a blinded pairwise evaluation protocol, using the underlying benign request as the reference. Responses are presented as A/B pairs with condition identities concealed and presentation order balanced.

Quality $Q \in \{1, \ldots, 5\}$ measures instruction fulfillment, relevance and correctness, and coherence. We report the mean paired change $\Delta Q = \text{mean}(Q_{\text{post}} - Q_{\text{base}})$; identical outputs contribute zero change. Over-refusal rate (ORR) is the fraction of all prompts in each input group for which a baseline non-refusal becomes a refusal after intervention, using denominators of 60 and 300 for plain and style-only inputs, respectively.

Confidence intervals use paired bootstrap resampling over benign-request identities, retaining all associated response pairs and wrapper variants. An input group passes when valid evaluation coverage

Table 19: Benign utility of frozen natural restoration. Active B/SO reports gate activations over the total numbers of plain-benign and style-only prompts. No intervention-induced refusals are observed, and $\Delta Q_B = 0$ for all configurations. Values for ΔQ_{SO} are means with 95% confidence intervals.

Model	Operator	Active B/SO	ΔQ_{SO} [95% CI]
Qwen2.5-7B	Gated natural	$0/60$ / $5/300$	$0.000\,[0,\ 0]$
	Component natural		$0.040\,[0.000,\ 0.120]$
Llama-3.1-8B	Gated natural	$1/60$ / $7/300$	$-0.003\,[-0.010,\ 0]$
	Component natural		$-0.003\,[-0.010,\ 0]$
Gemma-2-9B	Gated natural	$1/60$ / $14/300$	$0.000\,[0,\ 0]$
	Component natural		$0.000\,[0,\ 0]$

is at least 95% and

$$\mathrm{UCB}_{95\%}(\mathrm{ORR}) \leq 0.05, \qquad \mathrm{LCB}_{95\%}(\Delta Q) \geq -0.25.$$

Pooled utility preservation requires both the plain and pooled style-only groups to pass. Family-level outcomes are evaluated separately.

Table 19 summarizes natural-restoration utility. Both natural-restoration operators satisfy the pooled and family-level criteria across all three models. Targeted steering also satisfies the pooled criteria across models; the only family-level exception is the DSN subset for Llama.

E.6 Refusal-Direction Control

We examine whether direct refusal induction preserves the harmfulness score and benign task performance. Following Arditi et al. (2024), we construct a refusal direction from the difference between mean harmful and harmless calibration activations and freeze it after candidate selection. We add the selected vector at its chosen layer across all token positions throughout generation, with a fixed coefficient of $+1$ and no gating or H protection.

Refusal-direction addition reduces ASR to 2.33%, 0.00%, and 0.00% for Qwen, Llama, and Gemma, respectively (Table 20). The H score increases beyond the equivalence margin for Qwen and Gemma, while remaining within it for Llama. All three models exhibit substantial over-refusal and response-quality loss on both plain and wrapped benign requests.

Table 20: Refusal-direction control. Attack cohorts contain 472/88, 512/92, and 465/87 prompts/goals for Qwen, Llama, and Gemma, respectively. Utility evaluation uses 60 plain benign requests and 300 wrapped versions per model. Brackets denote 95% paired bootstrap confidence intervals, clustered by harmful goal for attack metrics and by benign identity for utility. H equivalence requires the entire normalized interval to lie within $[-1, 1]$.

(a) Attack suppression and H preservation

Model	Original ASR (%)	RD ASR (%)	$\Delta H/\epsilon_H$
Qwen	37.08	2.33 [0.85, 4.11]	1.757 [1.672, 1.844]
Llama	22.07	0.00 [0.00, 0.00]	−0.107 [−0.206, −0.005]
Gemma	35.91	0.00 [0.00, 0.00]	1.085 [1.028, 1.139]

(b) Benign utility

Model	Input	ORR (%)	ΔQ
Qwen	Plain	33.33 [21.67, 45.00]	−2.283 [−2.667, −1.900]
	Wrapped	67.00 [60.00, 73.67]	−2.813 [−3.047, −2.567]
Llama	Plain	88.33 [80.00, 96.67]	−3.050 [−3.333, −2.733]
	Wrapped	91.67 [86.33, 96.00]	−2.917 [−3.117, −2.690]
Gemma	Plain	58.33 [46.67, 70.00]	−2.917 [−3.167, −2.650]
	Wrapped	77.00 [70.00, 83.33]	−2.887 [−3.090, −2.650]

Zero observed successes yield degenerate empirical bootstrap intervals of $[0, 0]$.