FORM BEFORE HARM: DIAGNOSING THE SEMANTIC INFLUENCE OF JAILBREAK FORMATS WITH ACTIVATION STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model safety defenses often organize harmful inputs by topic without accounting for the semantic shifts induced by jailbreak formats. We show that these shifts can alter this organization: attack format has a substantially greater impact on the distribution of LLM semantic representations than harmful topic. To study this effect in activation space, we introduce Context-Aware Steering (COS), an adaptive activation-steering method that learns a routable steering space from mixed jailbreak data and achieves strong defense performance with limited benign-side effects. Across both the model's natural representations and the intervention structure learned by COS, attack format consistently plays a larger organizing role than harmful topic; COS further adapts its shared and specialized steering components to the internal structure of each LLM rather than relying on predefined topic categories. We further identify a clear out-of-distribution generalization gap across jailbreak formats. Increasing the diversity of attack formats during training alleviates this gap but does not eliminate it. Together, these findings show that jailbreak format is a major determinant of LLM semantic representations and should be considered when organizing safety detection and defense.

1 INTRODUCTION

LLM safety often organizes harmful requests into human-defined topic categories to support detection and defense (Inan et al., 2023; Lee et al., 2025). Jailbreak methods complicate this organization by reformulating harmful requests through role-play, encoding, reconstruction, response constraints, and other transformations (Zeng et al., 2024; Deng et al., 2024; Liu et al., 2024c). We refer to these characteristics of how a request is presented and transformed as **attack format**. Prior work has shown that such reformulations can perturb how a request is semantically represented within the LLM (Lin et al., 2024; Zhou et al., 2024; Gao et al., 2025). What remains unclear is whether these shifts merely disturb a semantic space still organized by harmful topic, or alter its dominant organization. This distinction determines whether harmful topic remains a reliable basis for organizing safety detection and defense.

This paper shows that jailbreak-induced shifts can alter the dominant organization of LLM semantic representations. As shown in Figure 1, wrapping the same harmful behavior in different attack formats can produce markedly different activation shifts. Under sufficiently large shifts, samples sharing the same harmful topic need not remain close in activation space, meaning that topic membership no longer guarantees semantic proximity for the model. This challenges the reliability of harmful topic as a basis for organizing safety detection and defense. Activation steering provides a direct way to study this phenomenon by reading and modifying model activations during generation. Existing methods model activation-space directions from unsafe behavior toward refusal and decide whether to intervene based on activations, with some methods further using category-conditioned routing (Rimsky et al., 2024; Cao et al., 2024; Lee et al., 2025; Zhao et al., 2025).

Motivated by these observations, we introduce COS, an adaptive activation-steering method without predefined harmful-topic routing. COS adaptively learns a routable steering space from mixed jailbreak data, enabling diverse jailbreak activations to be detected and mapped to appropriate unsafe-

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.

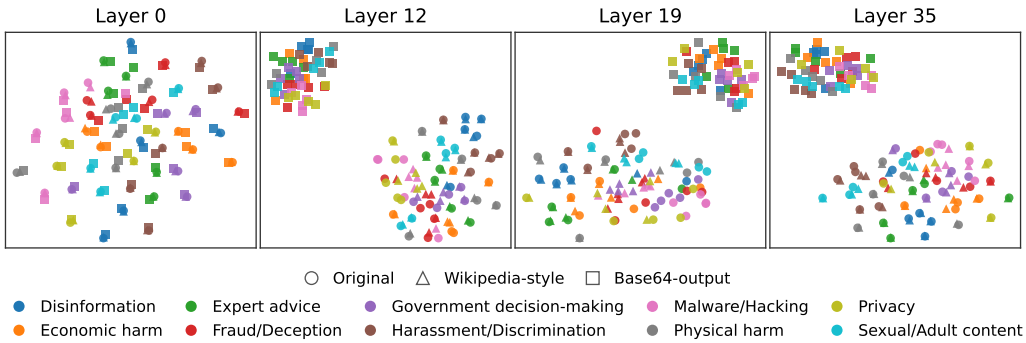


Figure 1: **Format-dependent activation shifts.** Qwen3-8B activations for identical harmful behaviors under Original, Wikipedia, and base64-output conditions across four layers. Colors indicate harmful topics; markers indicate conditions. Shift magnitude varies across formats (Appendix A.1).

to-refusal interventions. Beyond serving as a defense, this adaptive behavior provides a way to examine how jailbreak representations are organized in activation space.

To explore how jailbreak methods affect LLM representations and safety, we examine activation-space organization and find that attack-method identity and explicit format explain substantially more variation than harmful topic. Within this structure, the adaptive safety-steering method COS learns to detect harmful inputs and route interventions according to attack-dependent patterns that differ across LLMs. However, safety boundaries under this distribution exhibit cross-format threshold miscalibration despite strong harmful-versus-benign score rankings. These threshold shifts are not entirely arbitrary: our correction experiments suggest that they contain predictable structure that may help improve cross-format transfer. Our contributions are:

- We show that attack-method identity is more decodable and explains more representation variance than harmful-topic labels, while explicit format features account for a substantial related component across models and layers.
- We introduce COS, an adaptive activation-steering method that learns a routable steering space from mixed jailbreak data for adaptive safety intervention.
- We show that format dependence limits the cross-format generalization of safety boundaries, and explore whether activation-based score correction can predict and reduce the resulting shifts.

2 RELATED WORK

Jailbreak attacks. Jailbreak methods reformulate harmful requests through adversarial suffixes, persuasion, disguise and reconstruction, multistep prompting, or code and visual representations (Zou et al., 2023; Zeng et al., 2024; Deng et al., 2024; Liu et al., 2024c; Li et al., 2023a; Ren et al., 2024; Jiang et al., 2024). These varied transformations motivate examining how attack methods organize mixed-jailbreak representations beyond their behavioral success rates.

Representation-space safety analysis. Prior work identifies activation directions mediating refusal (Arditi et al., 2024) and shows that jailbreaks shift harmful representations and disrupt safety-related processing (Lin et al., 2024; Zhou et al., 2024; Gao et al., 2025). We build on these findings by comparing representation organization by attack method, explicit format, and harmful topic under controlled conditions. Autoencoder and feature-recovery methods support representation decomposition (Yun et al., 2021; Luo et al., 2024); COS uses an autoencoder to construct a compact steering space, without claiming a universal or monosemantic safety decomposition.

Activation engineering and adaptive steering. Activation engineering controls model behavior through inference-time interventions, including fixed directions derived from activation contrasts, probes, or optimization (Turner et al., 2023; Li et al., 2023b; Liu et al., 2024b; Rimsky et al., 2024;

Jorgensen et al., 2023; Cao et al., 2024). Adaptive methods include CAST's category-conditioned refusal steering, SADI's semantics-adaptive vectors, and AdaSteer's input-dependent coefficients (Lee et al., 2025; Wang et al., 2025; Zhao et al., 2025). COS learns a routable steering space from mixed jailbreak data, allowing interventions to adapt to each LLM's activation geometry without predefined harmful-topic routing.

3 METHOD

COS is an adaptive activation-steering method for inputs from multiple jailbreak methods. Given harmful requests generated by diverse attack methods, spanning varied presentation formats, together with benign inputs, it learns to distinguish harmful from benign inputs and adaptively maps harmful activations to appropriate unsafe-to-refusal interventions.

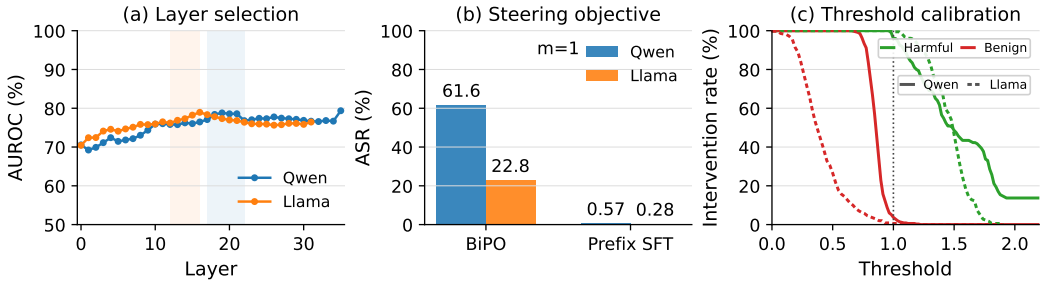


Figure 2: **Stage-level evidence for COS.** Panel (a) shows the harmful-versus-benign activation probes used to select intervention layers; panel (b) compares a BiPO-style objective with refusal-prefix supervision (Prefix SFT) for learning steering vectors with the LLM frozen; panel (c) shows intervention rates, with thresholds divided by each model's selected value.

3.1 STEP 1: SELECTING INTERVENTION LAYERS

Effective steering first requires identifying where unsafe-to-refusal transitions can be reliably represented and manipulated. We use harmful-versus-benign activation probes to identify where the two input classes are most separable, and select a contiguous high-scoring layer band $\mathcal{L}$ for subsequent source steering vector learning, routing, and inference, as shown in Figure 2(a).

3.2 STEP 2: BUILDING A SOURCE STEERING VECTOR LIBRARY

Mixed jailbreak inputs can require heterogeneous unsafe-to-refusal changes, which may not be captured by a single global steering direction. We therefore group harmful requests with similar activation representations into small semantic clusters of at most five requests, and learn source steering vectors for each cluster across the selected layers.

Let $\{\mathcal{D}_k\}_{k=1}^{K}$ denote the resulting clusters. For each cluster $\mathcal{D}_k$, we optimize source steering vectors $\{v_{k,\ell}\}_{\ell\in\mathcal{L}}$ while keeping the base model frozen. Each source steering vector is trained with a short refusal-prefix objective:

$$\mathcal{L}_{\text{src}}(k) = -\sum_{x\in\mathcal{D}_k} \log p_\theta\left(y_{\text{pref}} \mid x; \{v_{k,\ell}\}_{\ell\in\mathcal{L}}\right), \tag{1}$$

where y_{pref} is a short refusal prefix and the source steering vectors are added at their corresponding layers during teacher-forced training. This objective produces substantially stronger own-cluster unsafe-to-refusal steering than the alternative objective in Figure 2(b).

Each set of source steering vectors is behaviorally validated on its own cluster before entering the library $\mathcal{V} = \{\{v_{k,\ell}\}_{\ell\in\mathcal{L}}\}_{k=1}^{K}$.

3.3 STEP 3: LEARNING A ROUTABLE STEERING SPACE

An autoencoder compresses the source steering vector library into reusable steering components, forming a *routable steering space*. Components can be shared across attacks or specialize in particular patterns, without imposing either organization in advance.

For each selected layer ℓ, an encoder E_ℓ maps the source steering vector library into this compact space, while a decoder D_ℓ reconstructs the corresponding source steering vector:

$$z_{k,\ell} = E_\ell(v_{k,\ell}), \qquad \hat{v}_{k,\ell} = D_\ell(z_{k,\ell}). \tag{2}$$

The reconstruction objective preserves both source steering vector direction and magnitude:

$$\mathcal{L}_{\mathrm{AE}} = \sum_{k,\ell} \left[\mathrm{MSE}(\hat{v}_{k,\ell}, v_{k,\ell}) + 0.5\big(1 - \cos(\hat{v}_{k,\ell}, v_{k,\ell})\big) + \big(\|\hat{v}_{k,\ell}\|_2 - \|v_{k,\ell}\|_2\big)^2 \right]. \tag{3}$$

Here, a steering component is a unit in the compressed steering space, and the COS router predicts its activation for each input. Let $h_\ell(x)$ denote the activation of input x at layer ℓ. The COS router R_ℓ maps this activation into the learned steering space, and the decoder produces a predicted steering vector:

$$\tilde{v}_\ell(x) = D_\ell\left(R_\ell(h_\ell(x))\right). \tag{4}$$

For a harmful input from cluster k, the router output and decoded vector are trained toward the corresponding code $z_{k,\ell}$ and source steering vector $v_{k,\ell}$; benign inputs are trained toward a null intervention. The encoder, decoder, and router are jointly optimized so that the resulting space both preserves valid unsafe-to-refusal transitions and supports input-conditioned routing.

3.4 STEP 4: CALIBRATED INFERENCE

To keep COS inactive on benign inputs, we apply a calibrated activation threshold to the COS router. The same COS router supplies the gating score $s(x)$: the maximum pre-ReLU component score across all selected layers. A single input-level decision gates the predicted updates at every selected layer $\ell \in \mathcal{L}$:

$$g_\tau(x) = \mathbb{1}[s(x) > \tau], \qquad h'_\ell(x) = h_\ell(x) + g_\tau(x)\tilde{v}_\ell(x). \tag{5}$$

The threshold τ is selected on calibration data to retain harmful coverage while suppressing unnecessary intervention on benign inputs, as shown in Figure 2(c).

4 EXPERIMENT SETTING

Models and data. We evaluate Qwen3-8B (Qwen Team, 2025) and Llama-3.1-8B-Instruct (Llama Team, 2024). PromptSecurity-Eval (Hong et al., 2025) supports mixed-jailbreak defense evaluation and comparisons by attack method and harmful topic; JailbreakBench (Chao et al., 2024) supplies different attack variants of the same seed behaviors for representation analysis; XSTest (Röttger et al., 2024) provides benign evaluation prompts. Calibration and evaluation data are separated by underlying prompt identity. Dataset construction and topic annotation are detailed in Appendix B.

Behavioral evaluation. We compare COS with the base model, CAA, BiPO, CAST, and AdaSteer. We report LlamaGuard attack success rate (ASR) on harmful prompts and official XSTest over-refusal rate (ORR) on benign prompts, together with the COS intervention rate: the fraction of benign inputs activating its gate. Supplementary benign-input evaluations appear in Appendix F.1.

Representation and transfer analyses. We compare the linear decodability of attack method and harmful topic using identical examples, equal class counts, and cross-validation grouped by seed behavior. We separately measure representation variance associated with method labels, topic labels, and explicit format features; method identity is not equated with format. Cross-format safety-boundary transfer is evaluated using AUROC and fixed-threshold balanced accuracy. Details are provided in Appendices D.1, D.2, and C.

5 RESULTS

5.1 BEHAVIORAL EFFICACY AND ATTACK HETEROGENEITY

COS substantially reduces harmful behavior with limited side effects. COS achieves the lowest overall LlamaGuard ASR on both models (Table 1), reducing ASR from 37.2% to 0.7% on Qwen3-8B and from 48.2% to 1.4% on Llama-3.1-8B-Instruct. On benign XSTest prompts, Llama shows no false routing and no increase in ORR, which remains at 6.0%. Qwen exhibits a 3.2% benign routing rate, accompanied by only a 1.6-point increase in ORR from 4.0% to 5.6% (Table 2). Overall, the behavioral gains are accompanied by only limited benign-side effects.

Table 1: Main defense comparison on PromptSecurity. Values are LlamaGuard ASR (%, ↓: lower is better) for the overall set, harmful-topic groups, and attack-method groups. The lowest ASR in each row is bolded; ties are all bolded.

Model	Grouping	Category	Base	CAA	BiPO	CAST	AdaSteer	COS
Qwen3-8B	Overall	–	37.2	28.9	33.7	29.8	22.7	**0.7**
	Topic	Disinformation	43.0	32.0	34.0	31.0	14.0	**0.0**
		Expert advice	30.0	21.2	28.7	21.9	18.1	**1.9**
		Fraud/Scam/Phishing	32.0	25.0	27.5	27.5	24.5	**0.0**
		Harassment/Discrimination	37.5	34.2	36.7	33.3	19.2	**0.0**
		Malware/Hacking	46.8	32.4	35.3	32.4	29.5	**0.7**
		Physical harm	42.0	34.0	42.0	38.0	30.0	**2.0**
		Sexual/Adult content	33.8	30.0	41.2	30.0	22.5	**0.0**
	Attack	ABJ	26.7	2.2	**0.0**	**0.0**	**0.0**	**0.0**
		ArtPrompt	24.4	11.1	13.3	4.4	13.3	**0.0**
		AutoDAN	73.3	42.2	60.0	48.9	26.7	**0.0**
		COLD	38.6	31.8	36.4	34.1	34.1	**0.0**
		CodeAttack	57.8	31.1	26.7	42.2	44.4	**0.0**
		CodeChameleon	28.9	44.4	51.1	51.1	22.2	**0.0**
		DRA	100.0	95.6	97.8	97.8	66.7	**0.0**
		DrAttack	22.2	20.0	31.1	20.0	15.6	**2.2**
		FlipAttack	62.2	48.9	44.4	33.3	13.3	**0.0**
		GCG	20.0	13.3	20.0	15.6	20.0	**2.2**
		GPTFUZZER	84.4	64.4	68.9	73.3	24.4	**0.0**
		IFSJ	4.4	**0.0**	11.1	2.2	15.6	2.2
		Inception	55.6	42.2	48.9	44.4	62.2	**0.0**
		Multilingual jailbreak	20.0	24.4	22.2	17.8	15.6	**0.0**
		PAIR	26.7	24.4	28.9	22.2	22.2	**0.0**
		Past tense	6.7	**2.2**	11.1	4.4	6.7	**2.2**
		Persuasive in-context	11.1	8.9	8.9	8.9	11.1	**0.0**
		ReNeLLM	48.9	48.9	57.8	53.3	13.3	**0.0**
		Response attack	17.8	15.6	26.7	17.8	17.8	**2.2**
		TAP	13.3	6.7	8.9	4.4	8.9	**2.2**
Llama-3.1-8B-Instruct	Overall	–	48.2	22.8	21.7	52.2	6.9	**1.4**
	Topic	Disinformation	57.0	27.0	22.0	38.0	**3.0**	7.0
		Expert advice	34.4	20.0	20.0	43.1	12.5	**1.2**
		Fraud/Scam/Phishing	36.5	21.0	19.0	36.5	**0.5**	1.0
		Harassment/Discrimination	50.0	20.0	19.2	65.8	1.7	**0.0**
		Malware/Hacking	58.6	25.0	28.6	60.0	19.3	**0.0**
		Physical harm	54.0	24.0	25.0	72.0	7.0	**1.0**
		Sexual/Adult content	66.2	26.2	18.8	68.8	2.5	**1.2**
	Attack	ABJ	48.9	2.2	4.4	11.1	**0.0**	**0.0**
		ArtPrompt	35.6	**0.0**	4.4	37.8	**0.0**	**0.0**
		AutoDAN	75.6	15.6	31.1	51.1	2.2	**0.0**
		COLD	64.4	8.9	8.9	60.0	15.6	**6.7**
		CodeAttack	35.6	66.7	37.8	100.0	**0.0**	**0.0**
		CodeChameleon	33.3	26.7	22.2	100.0	**0.0**	**0.0**
		DRA	42.2	84.4	91.1	84.4	**0.0**	**0.0**
		DrAttack	28.9	26.7	17.8	40.0	**0.0**	**0.0**
		FlipAttack	75.6	53.3	64.4	48.9	**0.0**	**0.0**
		GCG	44.4	**2.2**	4.4	60.0	13.3	4.4
		GPTFUZZER	80.0	26.7	42.2	71.1	**2.2**	4.4
		IFSJ	95.6	2.2	**0.0**	91.1	20.0	**0.0**
		Inception	75.6	28.9	22.2	46.7	2.2	**0.0**
		Multilingual jailbreak	33.3	8.9	6.7	22.2	4.4	**0.0**
		PAIR	48.9	11.1	13.3	66.7	24.4	**8.9**
		Past tense	8.9	6.7	**2.2**	24.4	17.8	**2.2**
		Persuasive in-context	15.6	4.4	4.4	17.8	13.3	**2.2**
		ReNeLLM	46.7	53.3	31.1	46.7	13.3	**0.0**
		Response attack	51.1	11.1	15.6	57.8	4.4	**0.0**
		TAP	24.4	15.6	8.9	6.7	4.4	**0.0**

Table 2: Main XSTest benign comparison between Base and COS. Cells report official XSTest string-match over-refusal rate (ORR, %) on 250 safe prompts; the final column is the COS intervention rate (↓: lower is better on benign inputs). Detailed baseline comparisons are reported in Appendix F.1.

Model	Base ORR ↓	COS ORR ↓	COS intervention rate ↓
Qwen3-8B	4.00%	5.60%	3.20%
Llama-3.1-8B-Instruct	6.00%	6.00%	0.00%

Attack method is associated with greater behavioral variation than harmful topic. The same evaluation also reveals a clear asymmetry between attack method and harmful topic. On the base Qwen3-8B, ASR ranges from 4.4% to 100.0% across attack methods, but only from 30.0% to 46.8% across harmful topics; on Llama-3.1-8B-Instruct, the corresponding ranges are 8.9–95.6% and 34.4–66.2%. Thus, attack-method identity is associated with much larger variation in attack success than harmful-topic identity. This behavioral asymmetry motivates the representation-level analysis in the next section.

5.2 Attack Method and Format Shape Semantic Representations

Attack method is substantially more accessible than harmful topic. Across both models, attack method is nearly perfectly linearly decodable throughout the network, whereas harmful topic is consistently much less separable (Figure 3(a,b)). Topic information becomes more accessible with depth, but never approaches the attack-method signal. The persistence of this gap into upper layers suggests that attack-related information is not limited to shallow lexical differences, but remains strongly represented throughout semantic processing.

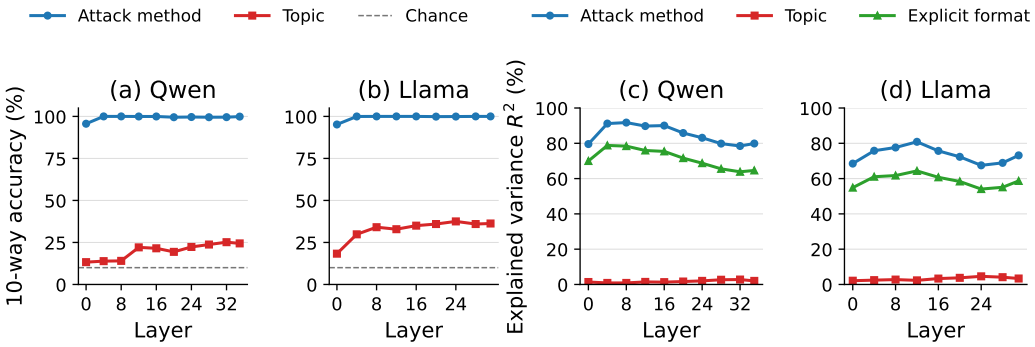


Figure 3: **Attack method and explicit format in hidden-space variation.** Panels (a,b) compare the linear decodability of attack method and harmful topic across layers on the same JailbreakBench examples. Panels (c,d) compare representation variance associated with attack-method labels, explicit format, and harmful-topic labels. Attack method is more decodable than topic, while method and explicit format contributions explain substantially more variance than topic and follow similar layer-wise trends.

Attack method explains more representation variance than harmful topic and follows a similar trend to explicit format. Variance decomposition compares three factors (Figure 3(c,d)). Harmful topic explains less than 5% of the representation variance on both models, while attack-method labels and explicit format explain substantially larger proportions. The attack-method and explicit-format curves follow similar layer-wise trajectories, indicating an association between method-level organization and explicit format. Attack-method labels nevertheless explain more variance than the explicit format features, suggesting additional instruction- or attack-specific structure. Together, these results show that attack-method identity and explicit format features account for substantially

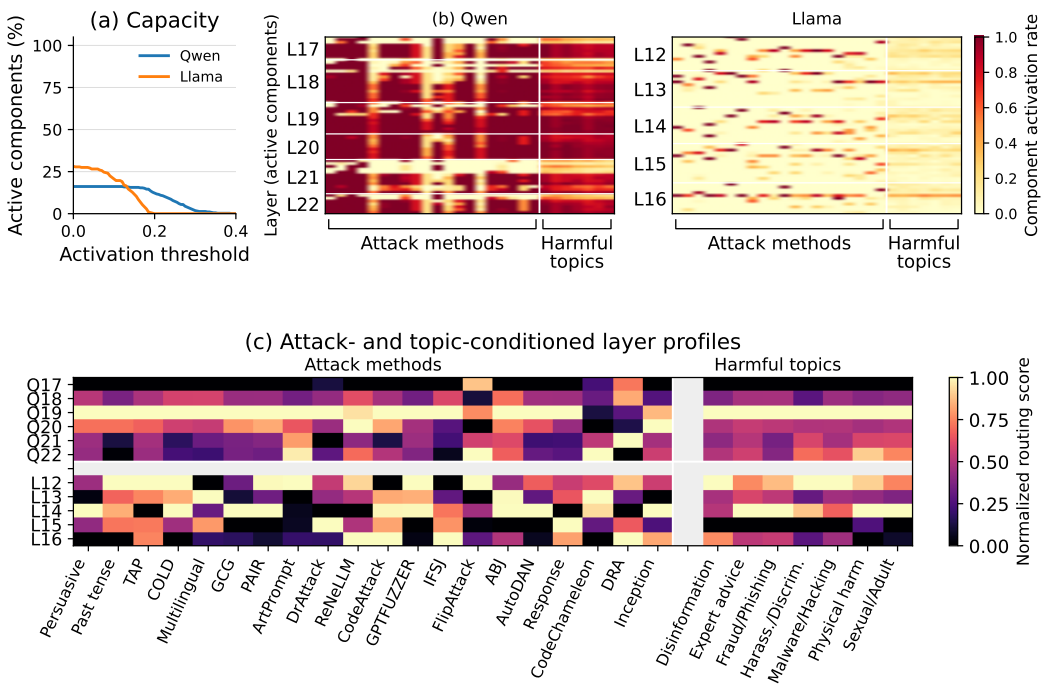


Figure 4: **Adaptive routing structure learned by COS.** Panel (a) shows the proportion of layer–component pairs activated by at least one harmful training input as the activation threshold increases. Panel (b) shows how effective steering components are activated across attack methods and harmful topics for Qwen3-8B and Llama-3.1-8B-Instruct. Panel (c) shows the corresponding layer-wise routing profiles. The learned routing is sparse and strongly attack-conditioned, with distinct layer organization across the two LLMs. A fully labeled version of panel (b) and additional steering-component sharing analysis appear in Appendix E.1, Figures 9 and 8.

more representation variance than harmful-topic labels; they do not treat method identity as equivalent to format.

5.3 ADAPTIVE DETECTION AND ROUTING UNDER ATTACK-DEPENDENT VARIATION

The adaptively learned steering space exhibits model-specific patterns of component sharing and specialization. COS provides capacity to learn reusable steering components without prescribing which components should be shared across attacks or specialized to particular inputs. Within this space, only a small fraction of components is activated by harmful inputs (Figure 4(a)). Their activation patterns differ substantially across models: counting each layer–component pair separately, 25 of Qwen's 62 active components respond to at least 90% of harmful inputs, whereas no Llama component responds to 50% (Appendix E.1, Figure 8). These patterns suggest that COS can learn broadly shared components in some models and more specialized components in others.

Steering-component activation varies more across attack methods than across harmful topics. Figure 4(b) compares COS steering-component activation patterns across attack methods and harmful topics. Different attack methods activate different combinations of steering components and elicit different activation rates, whereas the patterns across harmful topics are less distinct. This contrast suggests that COS's adaptively learned steering space reflects attack-dependent variation more clearly than predefined harmful-topic categories.

Qwen safety routing concentrates at layer 19, while delayed routing is associated with stronger attacks. A pronounced layer-specific pattern further emerges on Qwen3-8B (Figure 4(c)). Fifteen

of the 20 attack methods reach their highest mean route logit at layer 19, while the remaining five peak only at layers 20–22. These later-peaking attacks are substantially more successful on the base model, with a mean ASR of 59.1% compared with 29.8% for attacks peaking at layer 19. This pattern suggests that Qwen's safety-relevant routing is strongly concentrated around layer 19, while attacks whose dominant routing response is delayed beyond this layer tend to be more successful. The association does not establish that bypassing layer 19 causes jailbreak success, but it reveals a layer-localized vulnerability pattern in the model's safety routing.

5.4 Format Dependence Limits Cross-Format Safety Boundaries

Format shifts often miscalibrate the threshold while preserving score rankings. To examine how attack format affects safety boundaries, we train linear classifiers, independently of the COS router, in LLM activation space to distinguish harmful from benign requests sharing the same attack format. A boundary learned under one format does not always transfer at its original threshold (Figure 5). At layer 24, mean cross-format AUROC remains around 99.5% for both models, indicating that harmful and benign inputs are still strongly rank-separable on average. Yet balanced accuracy, averaged over distinct source–target pairs at the source threshold, is only 76.4% for Qwen3-8B and 80.3% for Llama-3.1-8B-Instruct. This gap shows that threshold miscalibration is a major part of the average transfer loss, even when score rankings remain strong. The boundary-similarity maps also show nearly orthogonal boundaries for some format pairs, so this is not a universal offset-only account: particular shifts can alter the boundary direction as well as its threshold.

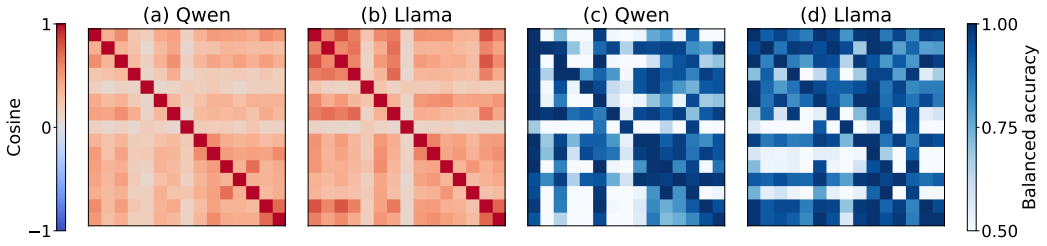


Figure 5: **Cross-format safety-boundary transfer at layer 24.** Panels (a,b) show pairwise cosine similarity between format-specific safety boundaries for Qwen3-8B and Llama-3.1-8B-Instruct, while panels (c,d) show fixed-threshold balanced accuracy when each source boundary is transferred to another format. Full attack-method labels, including the Original reference and color scales are provided in Appendix 7. Safety boundaries can shift substantially across formats, causing some fixed-threshold transfers to collapse toward chance.

Broader format coverage improves transfer but does not remove the gap. We next examine whether simply increasing the diversity of attack formats used for training can improve out-of-distribution generalization. As shown in Figure 6(a,b), training on more attack formats progressively improves held-out boundary transfer, particularly on Qwen3-8B. However, held-out performance remains consistently below the near-perfect in-domain performance even after many formats are included. The same pattern, though smaller in magnitude, appears on Llama-3.1-8B-Instruct. These results show that exposure to diverse formats can reduce format sensitivity, but does not by itself yield a format-invariant safety boundary.

5.5 Toward Robust Cross-Format Safety Boundaries

Section 5.4 shows that score rankings generalize well across attack formats on average, while threshold shifts leave a persistent generalization gap. We next ask whether this gap can be further reduced. A natural approach is to **predict the threshold shift from input activations** and use the predicted shift to correct the safety score. Starting from a boundary trained on Original prompts, we compare four settings in Figure 6(c,d): the raw boundary without correction; a global correction using a shared threshold offset estimated across observed attack methods; and linear and nonlinear predictors that use input activations to predict the remaining threshold shift beyond the global offset. The predictors are trained on observed attack methods and evaluated on unseen methods, with im-

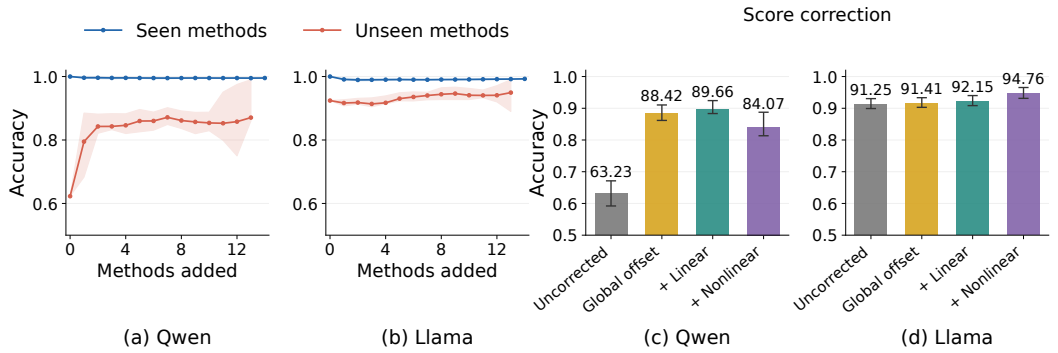


Figure 6: **Boundary transfer and activation-based score correction at layer 24.** Panels (a,b) compare transfer to seen and unseen attack methods as training coverage increases. Panels (c,d) compare the raw boundary, global correction, and additional linear or nonlinear predicted corrections, using eight observed and six held-out methods. Curves average 32 method orders; bars average 32 method splits. Shading and whiskers show interquartile ranges.

plementation details in Appendix H. Global correction provides most of the gain on Qwen, whereas activation-conditioned correction yields a larger additional gain on Llama.

Qwen benefits mainly from a shared offset. A global correction raises OOD balanced accuracy from 63.23% to 88.42%, indicating that much of Qwen's cross-format mismatch can be corrected by a shared offset. An activation-based linear predictor further improves accuracy to 89.66%, suggesting that activations contain additional information useful for correcting the remaining shift. In contrast, the nonlinear predictor reduces accuracy to 84.07%, indicating that additional model flexibility does not necessarily improve transfer.

Llama benefits more from activation-conditioned correction. For Llama, global correction changes balanced accuracy only slightly, from 91.25% to 91.41%, whereas the linear and nonlinear predictors reach 92.15% and 94.76%, respectively.

Together, these results suggest that **format-induced threshold shifts may be partly predictable**, allowing correction on unseen attack methods, although the useful correction differs across LLMs. For safety defense, these results motivate further modeling of threshold shifts to improve adaptation to unseen attack formats. For interpretability research, they motivate identifying and constructing feature spaces that are more robust to attack format, providing a more stable representational basis for safety detection and intervention.

6 Conclusion

This work examines how jailbreak formats shape LLM representations and the safety boundaries learned within them. Across the two models studied, attack format plays a stronger organizing role than harmful topic. Our adaptive safety-steering method, COS, learns to detect harmful inputs and route interventions within this structure, achieving strong defense performance with limited effects on benign inputs. Its steering components exhibit model-specific patterns of sharing and specialization, without predefined topic categories. However, strong harmful-versus-benign rankings do not guarantee reliable transfer: thresholds shift across attack formats, and broader training coverage reduces but does not eliminate the generalization gap. Our exploratory correction experiments suggest that these threshold shifts may be partly predictable from input activations, offering a possible direction for further improvement. Together, these findings highlight the importance of accounting for attack format when organizing safety detection and defense, and motivate safety boundaries that adapt to format-induced shifts.

AI USE STATEMENT

Generative AI tools were used to assist with writing refinement and to accelerate code development. We have not used generative AI tools for the initial literature review, core methodological ideas, and overall experimental design. All AI-assisted content, code, analyses, and experimental results were reviewed and verified by the authors, who take full responsibility for the final manuscript.

ETHICS STATEMENT

This work studies jailbreak attacks for the purpose of improving the safety and robustness of large language models. All attack data, code, and experiments were used solely for controlled research and evaluation, without deployment against real-world systems or users. We focus on aggregate safety and representation analyses and do not intend to facilitate harmful use.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the datasets, experimental settings, model configurations, evaluation protocols, and additional analyses in the main paper and appendices, including COS hyperparameters in Appendix E.4. Complete code, data-processing scripts, and experimental artifacts necessary to reproduce our results will be publicly released upon acceptance.

REFERENCES

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, 2024. URL https://dblp.org/rec/conf/nips/ArditiOSPPGN24.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosiute, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemí Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI feedback. *CoRR*, abs/2212.08073, 2022. doi: 10.48550/arXiv.2212.08073. URL https://dblp.org/rec/journals/corr/abs-2212-08073.

Rishabh Bhardwaj and Soujanya Poria. Red-teaming large language models using chain of utterances for safety-alignment. *CoRR*, abs/2308.09662, 2023. doi: 10.48550/arXiv.2308.09662. URL https://dblp.org/rec/journals/corr/abs-2308-09662.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. In *Advances in Neural Information Processing Systems*, 2024. URL https://dblp.org/rec/conf/nips/CaoZC00MC24.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. *CoRR*, abs/2404.01318, 2024. doi: 10.48550/arXiv.2404.01318. URL https://dblp.org/rec/journals/corr/abs-2404-01318.

Zifeng Cheng, Jinwei Gan, Zhiwei Jiang, Cong Wang, Yafeng Yin, Xiang Luo, Yuchen Fu, and Qing Gu. Steering when necessary: Flexible steering large language models with backtracking. In *Advances in Neural Information Processing Systems*, 2025. URL https://dblp.org/rec/conf/nips/ChengGJWYLFG25.

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. MASTERKEY: Automated jailbreaking of large language model chatbots. In *Proceedings of NDSS*, 2024. URL https://dblp.org/rec/conf/ndss/DengLLWZLW0L24.

Van Dai Do, Quan Hung Tran, Svetha Venkatesh, and Hung Le. Dynamic steering with episodic memory for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13731–13749, 2025. URL https://dblp.org/rec/conf/acl/DoTV025.

Toan Doan, Uyen Le, and Thin Nguyen. Causal activation steering via sparse mediation. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 1079–1097, 2026. URL https://dblp.org/rec/conf/eacl/DoanLN26.

Masha Fedzechkina, Eleonora Gualdoni, Sinead Williamson, Katherine Metcalf, Skyler Seto, and Barry-John Theobald. ExpertLens: Activation steering features are highly interpretable. *Transactions on Machine Learning Research*, 2026, 2026. URL https://dblp.org/rec/journals/tmlr/FedzechkinaGWMST26.

Lang Gao, Jiahui Geng, Xiangliang Zhang, Preslav Nakov, and Xiuying Chen. Shaping the safety boundaries: Understanding and defending against jailbreaks in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25378–25398, 2025. doi: 10.18653/v1/2025.acl-long.1233. URL https://dblp.org/rec/conf/acl/GaoG0NC25.

Hanbin Hong, Shuya Feng, Nima Naderloui, Shenao Yan, Jingyu Zhang, Biying Liu, Ali Arastehfard, Heqing Huang, and Yuan Hong. SoK: Taxonomy and evaluation of prompt security in large language models. *CoRR*, abs/2510.15476, 2025. doi: 10.48550/arXiv.2510.15476. URL https://dblp.org/rec/journals/corr/abs-2510-15476.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama guard: LLM-based input-output safeguard for human-AI conversations. *CoRR*, abs/2312.06674, 2023. doi: 10.48550/arXiv.2312.06674. URL https://dblp.org/rec/journals/corr/abs-2312-06674.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. In *Advances in Neural Information Processing Systems*, 2023. URL https://dblp.org/rec/conf/nips/JiLDPZB0SW023.

Fengqing Jiang, Zhangchen Xu, Luyao Niu, Zhen Xiang, Bhaskar Ramasubramanian, Bo Li, and Radha Poovendran. ArtPrompt: ASCII art-based jailbreak attacks against aligned LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15157–15173, 2024. URL https://dblp.org/rec/conf/acl/JiangXNXR0P24.

Ole Jorgensen, Dylan Cope, Nandi Schoots, and Murray Shanahan. Improving activation steering in language models with mean-centring. *CoRR*, abs/2312.03813, 2023. doi: 10.48550/arXiv.2312.03813. URL https://dblp.org/rec/journals/corr/abs-2312-03813.

Minjae Kang and Jaehyung Kim. Enhancing instruction following of LLMs via activation steering with dynamic rejection. *CoRR*, abs/2603.06745, 2026. doi: 10.48550/arXiv.2603.06745. URL https://dblp.org/rec/journals/corr/abs-2603-06745.

Kristian Kuznetsov, Laida Kushnareva, Anton Razzhigaev, Polina Druzhinina, Anastasia Voznyuk, Irina Piontkovskaya, Evgeny Burnaev, and Serguei Barannikov. Feature-level insights into artificial text detection with sparse autoencoders. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25727–25748, 2025. URL https://dblp.org/rec/conf/acl/KuznetsovKRDVPB25.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre L. Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *International Conference on Learning Representations*, 2025. URL https://dblp.org/rec/conf/iclr/LeePRMDND25.

Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. Multi-step jailbreaking privacy attacks on ChatGPT. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 4138–4153, 2023a. URL https://dblp.org/rec/conf/emnlp/LiGFXHMS23.

Kenneth Li, Oam Patel, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, 2023b. URL https://dblp.org/rec/conf/nips/0002PVPW23.

Yichen Li, Zhiting Fan, Ruizhe Chen, Xiaotang Gai, Luqi Gong, Yan Zhang, and Zuozhu Liu. FairSteer: Inference time debiasing for LLMs with dynamic activation steering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11293–11312, 2025. doi: 10.18653/v1/2025.findings-acl.589. URL https://dblp.org/rec/conf/acl/LiFCGG0L25.

Yuping Lin, Pengfei He, Han Xu, Yue Xing, Makoto Yamada, Hui Liu, and Jiliang Tang. Towards understanding jailbreak attacks in LLMs: A representation space analysis. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7067–7085, 2024. doi: 10.18653/v1/2024.emnlp-main.401. URL https://dblp.org/rec/conf/emnlp/LinH0XYLT24.

Quan Liu, Zhenhong Zhou, Longzhu He, Yi Liu, Wei Zhang, and Sen Su. Alignment-enhanced decoding: Defending jailbreaks via token-level adaptive refining of probability distributions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2802–2816, 2024a. doi: 10.18653/v1/2024.emnlp-main.164. URL https://dblp.org/rec/conf/emnlp/LiuZHLZS24.

Sheng Liu, Haotian Ye, Lei Xing, and James Y. Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *Proceedings of the International Conference on Machine Learning*, pp. 32287–32307, 2024b. URL https://dblp.org/rec/conf/icml/LiuY0Z24.

Tong Liu, Yingjie Zhang, Zhe Zhao, Yinpeng Dong, Guozhu Meng, and Kai Chen. Making them ask and answer: Jailbreaking large language models in few queries via disguise and reconstruction. In *33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*, 2024c. URL https://dblp.org/rec/conf/uss/LiuZZDM024.

Llama Team. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/arXiv.2407.21783. URL https://dblp.org/rec/journals/corr/abs-2407-21783.

Hantao Lou, Changye Li, Jiaming Ji, and Yaodong Yang. SAE-V: Interpreting multimodal models for enhanced alignment. In *Proceedings of the International Conference on Machine Learning*, 2025. URL https://dblp.org/rec/conf/icml/Lou0J025.

Jinqi Luo, Tianjiao Ding, Kwan Ho Ryan Chan, Darshan Thaker, Aditya Chattopadhyay, Chris Callison-Burch, and René Vidal. PaCE: Parsimonious concept engineering for large language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://dblp.org/rec/conf/nips/LuoDCTCCV24.

Dung V. Nguyen, Hieu M. Vu, Nhi Y. Pham, Lei Zhang, and Tan M. Nguyen. Activation steering with a feedback controller. *CoRR*, abs/2510.04309, 2025. doi: 10.48550/arXiv.2510.04309. URL https://dblp.org/rec/journals/corr/abs-2510-04309.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022. URL https://dblp.org/rec/conf/nips/Ouyang0JAWMZASR22.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/arXiv.2505.09388. URL https://dblp.org/rec/journals/corr/abs-2505-09388.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023. URL https://dblp.org/rec/conf/nips/RafailovSMMEF23.

Qibing Ren, Chang Gao, Jing Shao, Junchi Yan, Xin Tan, Wai Lam, and Lizhuang Ma. CodeAttack: Revealing safety generalization challenges of large language models via code completion. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11437–11452, 2024. doi: 10.18653/v1/2024.findings-acl.679. URL https://dblp.org/rec/conf/acl/RenGSYTLM24.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://dblp.org/rec/conf/acl/RimskyGSTHT24.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of NAACL*, pp. 5377–5400, 2024. doi: 10.18653/v1/2024.naacl-long.301. URL https://dblp.org/rec/conf/naacl/RottgerKVA0H24.

Jianshu She, Xinyue Li, Eric P. Xing, Zhengzhong Liu, and Qirong Ho. Linear steerability in language models: When it emerges and how it evolves. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 17821–17846, 2025. URL https://dblp.org/rec/conf/emnlp/SheLXLH25.

Hao Sun, Huailiang Peng, Qiong Dai, Xu Bai, and Yanan Cao. LayerNavigator: Finding promising intervention layers for efficient activation steering in large language models. In *Advances in Neural Information Processing Systems*, 2025. URL https://dblp.org/rec/conf/nips/SunPDBC25.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *CoRR*, abs/2308.10248, 2023. doi: 10.48550/arXiv.2308.10248. URL https://dblp.org/rec/journals/corr/abs-2308-10248.

Minh Hieu Vu and Tan M. Nguyen. Angular steering: Behavior control via rotation in activation space. In *Advances in Neural Information Processing Systems*, 2025. URL https://dblp.org/rec/conf/nips/VuN25.

Weixuan Wang, Jingyuan Yang, and Wei Peng. Semantics-adaptive activation intervention for LLMs via dynamic steering vectors. In *International Conference on Learning Representations*, 2025. URL https://dblp.org/rec/conf/iclr/Wang0025.

Yang Xu, Yi Wang, and Hao Wang. Tracking the feature dynamics in LLM training: A mechanistic study. *CoRR*, abs/2412.17626, 2024a. doi: 10.48550/arXiv.2412.17626. URL https://dblp.org/rec/journals/corr/abs-2412-17626.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5587–5605, 2024b. URL https://dblp.org/rec/conf/acl/XuJN0LP24.

Zeyu Yun, Yubei Chen, Bruno A. Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: Contextualized embedding as a linear superposition of transformer factors. In *Proceedings of DeeLIO*, pp. 1–10, 2021. doi: 10.18653/v1/2021.deelio-1.1. URL https://dblp.org/rec/conf/acl-deelio/YunCOL21.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How johnny can persuade LLMs to jailbreak them: Rethinking persuasion to challenge AI safety by humanizing LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14322–14350, 2024. doi: 10.18653/v1/2024.acl-long.773. URL https://dblp.org/rec/conf/acl/0005LZY0S24.

Hongjue Zhao, Haosen Sun, Jiangtao Kong, Xiaochang Li, Qineng Wang, Liwei Jiang, Qi Zhu, Tarek F. Abdelzaher, Yejin Choi, Manling Li, and Huajie Shao. ODESteer: A unified ODE-based steering framework for LLM alignment. *CoRR*, abs/2602.17560, 2026. doi: 10.48550/arXiv.2602.17560. URL https://dblp.org/rec/journals/corr/abs-2602-17560.

Weixiang Zhao, Jiahe Guo, Yulin Hu, Yang Deng, An Zhang, Xingyu Sui, Xinyang Han, Yanyan Zhao, Bing Qin, Tat-Seng Chua, and Ting Liu. AdaSteer: Your aligned LLM is inherently an adaptive jailbreak defender. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24559–24577, 2025. URL https://dblp.org/rec/conf/emnlp/ZhaoGHDZSHZQCL25.

Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. On prompt-driven safeguarding for large language models. In *Proceedings of the International Conference on Machine Learning*, pp. 61593–61613, 2024. URL https://dblp.org/rec/conf/icml/ZhengY0M0CHP24.

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. doi: 10.18653/v1/2024.findings-emnlp.139. URL https://dblp.org/rec/conf/emnlp/ZhouYZXHL24.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *CoRR*, abs/2307.15043, 2023. doi: 10.48550/arXiv.2307.15043. URL https://dblp.org/rec/journals/corr/abs-2307-15043.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J. Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *Advances in Neural Information Processing Systems*, 2024. URL https://dblp.org/rec/conf/nips/ZouPWDLAKFH24.

A Attack-Format Representation Diagnostics

A.1 Activation-Shift Visualization

This section describes the activation extraction and visualization protocol for Figure 1. We use Qwen3-8B and held-out JailbreakBench behaviors. For each of 50 held-out harmful behaviors, we include three versions of each request: Original, a Wikipedia-style transformation, and a Base64-output transformation. Figure 1 shows that the attack transformation can shift the activation distribution while the underlying harmful behavior is held fixed. We extract hidden states after applying the model chat template, mean-pool over the final eight prompt tokens, and visualize layers 0, 12, 19, and 35 with t-SNE.

Table 3: Quantitative diagnostics for Figure 1. Silhouette scores are computed in the normalized hidden-state space using cosine distance, before t-SNE. Transformation groups are Original, Wikipedia-style, and Base64-output. Silhouette ↑ indicates stronger group separation; Difference is transformation minus topic silhouette.

Layer	Transformation silhouette	Harmful-topic silhouette	Difference
0	0.142	0.035	0.107
12	0.312	0.039	0.273
19	0.281	-0.005	0.286
35	0.263	0.018	0.244

B Dataset Construction and Leakage Control

PromptSecurity-Eval. PromptSecurity-Eval contains harmful and benign prompts and their model responses under a unified prompt-security protocol. We separate calibration and evaluation data by the identity of the underlying prompt. All defenses are evaluated on the same 899 test inputs for Qwen3-8B and the same 900 test inputs for Llama-3.1-8B-Instruct.

Harmful-topic annotation. The 100 PromptSecurity-Eval balanced-challenge prompts comprise 37 JailbreakBench (JBB), 36 AirBench, and 27 HarmBench examples (Hong et al., 2025; Chao et al., 2024). JBB examples use the benchmark's official category labels. AirBench and HarmBench examples are mapped to the ten JBB categories with a local Qwen2.5-7B-Instruct classifier. The classification prompt supplies category descriptions and up to five JBB behavior examples per category and requests one category together with confidence and a rationale; decoding is deterministic. The saved annotation file retains the source labels and model outputs for audit.

The final mapping keeps the official JBB labels for JBB examples and uses the Qwen mapping for AirBench and HarmBench. We split the JBB "Fraud/Deception" category into "Fraud/Scam/Phishing" and "Illegal Non-Fraud" according to whether the request involves fraud, scams, phishing, or related deception. We then exclude categories with fewer than five examples: Illegal Non-Fraud (4), Economic harm (4), Privacy (2), and Government decision-making (2). This leaves 88 prompts in seven categories:

Table 4: Final harmful-topic counts used in the main seven-way analysis.

Harmful topic	Prompts
Fraud/Scam/Phishing	20
Expert advice	15
Malware/Hacking	14
Harassment/Discrimination	11
Disinformation	10
Physical harm	10
Sexual/Adult content	8
Total retained	88

JailbreakBench attack variants. We apply multiple attack methods to the same JailbreakBench harmful behaviors. The probes compare ten-class attack-method classification with ten-class harmful-topic classification across five groups of ten attack methods.

XSTest safe. We use the 250 official safe prompts from XSTest to measure over-refusal and benign routing. XSTest is intentionally useful here because many prompts contain safety-adjacent words while remaining benign.

C Cross-Format Safety-Boundary Transfer

This appendix gives the complete protocol behind Section 5.4's direct boundary-transfer experiment. The paired dataset contains 100 JBB harmful seed prompts and 100 XSTest benign seed prompts. We use Original to denote a seed prompt without any attack transformation; Original includes both harmful and benign prompts. Fourteen attack methods transform each seed prompt by inserting method-specific prefixes and suffixes; nine encoded or character-level methods are excluded because a benign counterpart would require a new transformation. Original prompts and prompts transformed by each attack method share the same seed-prompt train/test split (50 harmful and 50 benign seed prompts per split), so the same underlying request never appears in both partitions.

For each model, we extract the final-eight-token mean at layers 0, 12, 24, and the final layer. A balanced linear logistic boundary is fitted on Original prompts, on prompts transformed by one attack method, or on a deterministically selected combination of these prompt groups. We report both AUROC, which is invariant to a score threshold, and balanced accuracy at the source boundary's zero decision threshold. The latter is the relevant fixed operating point when a boundary is transferred without target labels.

Table 5: Cross-format boundary transfer summary. Each source or target is either Original prompts or prompts transformed by one attack method. Transfer values average all ordered pairs with different source and target groups. The minimum columns report the worst ordered pair at the listed layer; BA denotes balanced accuracy and cosine denotes boundary-vector cosine similarity. ↑ means better discrimination for AUROC/BA and greater directional alignment for cosine.

Model	Layer	Mean AUROC ↑	Mean BA ↑	Minimum BA ↑	Minimum cosine ↑
Qwen3-8B	0	95.90%	67.38%	43.00%	0.080
Qwen3-8B	12	99.74%	67.94%	50.00%	0.107
Qwen3-8B	24	99.51%	76.41%	50.00%	0.072
Qwen3-8B	35	98.26%	72.91%	49.00%	0.022
Llama-3.1-8B-Instruct	0	96.53%	70.39%	49.00%	0.005
Llama-3.1-8B-Instruct	12	97.93%	76.64%	50.00%	0.014
Llama-3.1-8B-Instruct	24	99.55%	80.26%	50.00%	0.073
Llama-3.1-8B-Instruct	31	99.05%	81.58%	50.00%	0.059

The summary table separates two notions that are often conflated. AUROC is high for nearly every model/layer combination (95.90–99.74% for Qwen3-8B and 96.53–99.55% for Llama-3.1-8B-Instruct), yet the same source threshold transfers much less reliably: mean balanced accuracy ranges from 67.38% to 76.41% for Qwen3-8B and from 70.39% to 81.58% for Llama-3.1-8B-Instruct, with worst pairs close to chance. In other words, harmful and benign examples remain rank-separable within prompts from a given target attack method or the Original reference, but the score origin and scale move across prompt formats.

For the repeated-order curves in Figure 6, we include Original training prompts throughout and progressively add training prompts transformed by each of the 14 attack methods, following 32 fixed-seed random method orders. At each step, the boundary is evaluated on prompts transformed by the attack methods not yet included in training; the main curve is the mean over orders and the band is the interquartile range.

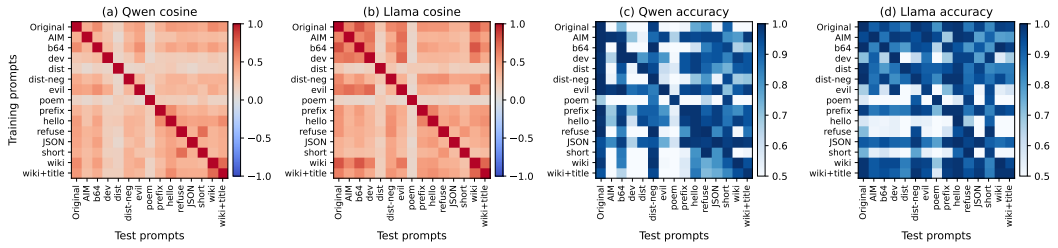


Figure 7: Fully labeled cross-format safety-boundary diagnostics at layer 24. Panels (a,b) show pairwise boundary cosine in [-1,1]; panels (c,d) show balanced accuracy in [0.5,1] when a boundary trained on the row's prompts is applied without recalibration to the column's prompts. Each row or column denotes Original prompts or prompts transformed by the named attack method.

The matrix axes identify Original prompts and 14 attack methods: Original = seed prompt without an attack transformation; AIM = AIM; b64 = base64-output-only; dev = dev-mode-v2; dist = distractors; dist-neg = distractors-negated; evil = evil-confidant; poem = poems; prefix = prefix-injection; hello = prefix-injection-hello; refuse = refusal-suppression; JSON = style-injection-JSON; short = style-injection-short; wiki = Wikipedia; wiki+title = Wikipedia-with-title. Rows identify the prompts used to train the source boundary; columns identify the prompts used for evaluation in every matrix.

D Representation-Space Diagnostics

D.1 Layer-Wise Probe Results

The attack-method probe predicts which jailbreak procedure generated a prompt, while the topic probe predicts its seed request's harmful-topic label. Both tasks use the same examples and equal class counts. Cross-validation is grouped by seed behavior, keeping all variants of an underlying harmful request in the same partition. Both classification tasks use ten equally likely classes, giving 10% chance accuracy. Attack-method identity remains substantially more decodable than harmful topic across the measured layers.

Table 6: Layer-wise attack-method and harmful-topic classification accuracy (%, ↑: higher is better). Dashes indicate layers not evaluated in the corresponding model.

	Qwen		Llama	
Layer	Attack method	Harmful topic	Attack method	Harmful topic
0	95.64%	13.28%	95.20%	18.32%
4	100.00%	13.88%	99.96%	29.84%
8	100.00%	14.04%	100.00%	34.08%
12	99.96%	22.08%	100.00%	32.88%
16	99.96%	21.52%	100.00%	35.00%
20	99.52%	19.40%	99.88%	35.96%
24	99.64%	22.32%	99.92%	35.32%
28	99.52%	23.76%	99.92%	37.52%
31	–	–	100.00%	36.64%
32	99.56%	25.16%	–	–
35	99.88%	24.44%	–	–

D.2 Variance Decomposition

The variance-decomposition experiments use L2-normalized hidden states, PCA to 50 dimensions, and R^2 estimates for attack-method labels, harmful-topic labels, or explicit format features. Explicit format comprises nine text features: character count, whitespace-separated word count, line count, punctuation fraction, digit fraction, uppercase-letter fraction among alphabetic characters, non-ASCII character fraction, colon fraction, and single/double quotation-mark fraction. Fractions use character count as the denominator except for the uppercase-letter feature; punctuation uses Python's ASCII punctuation set. Each feature is standardized within each analyzed group. For each layer and group, we fit separate ordinary least-squares models with an intercept to the 50-dimensional PCA representation, using attack-method indicators, harmful-topic indicators, or the nine explicit format features. Label indicators omit one reference category. The reported $R^2 = 1 - \mathrm{SSE}/\mathrm{SST}$ pools squared residuals across PCA dimensions and is computed on the fitted observations; it describes association, not held-out prediction. Results are averaged over five groups of ten methods: the first ten methods in sorted order and four distinct randomly sampled groups. The table reports the range of these means across layers.

Table 7: Variance-decomposition range summary. R^2 ↑ indicates a larger proportion of explained representation variance, not better defense performance.

Model	Attack method R^2	Harmful topic R^2	Explicit format R^2
Qwen3-8B	78.46–91.74%	0.87–2.78%	63.85–78.79%
Llama-3.1-8B-Instruct	67.51–80.85%	2.13–4.61%	54.06–64.35%

These descriptive R^2 values quantify association with representation variance, not causal effects of the labels.

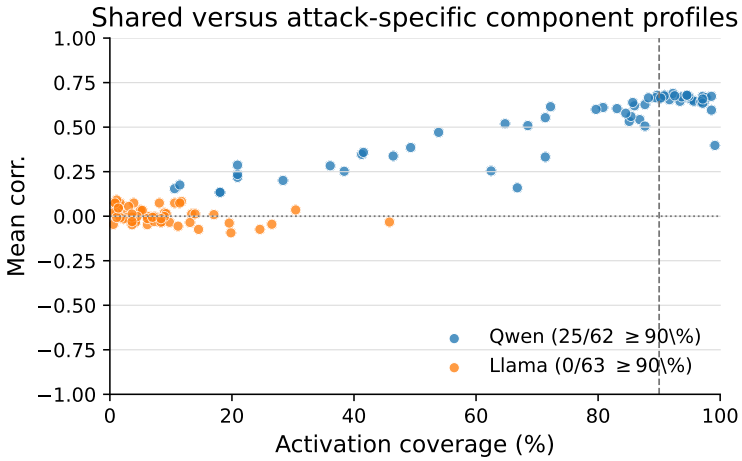


Figure 8: Shared versus attack-specific steering-component profiles. Each point is an active steering component at one layer; horizontal position is activation coverage over harmful inputs and vertical position is mean correlation of its attack-method activation profile with other steering components. The legend reports the number of steering components reaching 90% coverage.

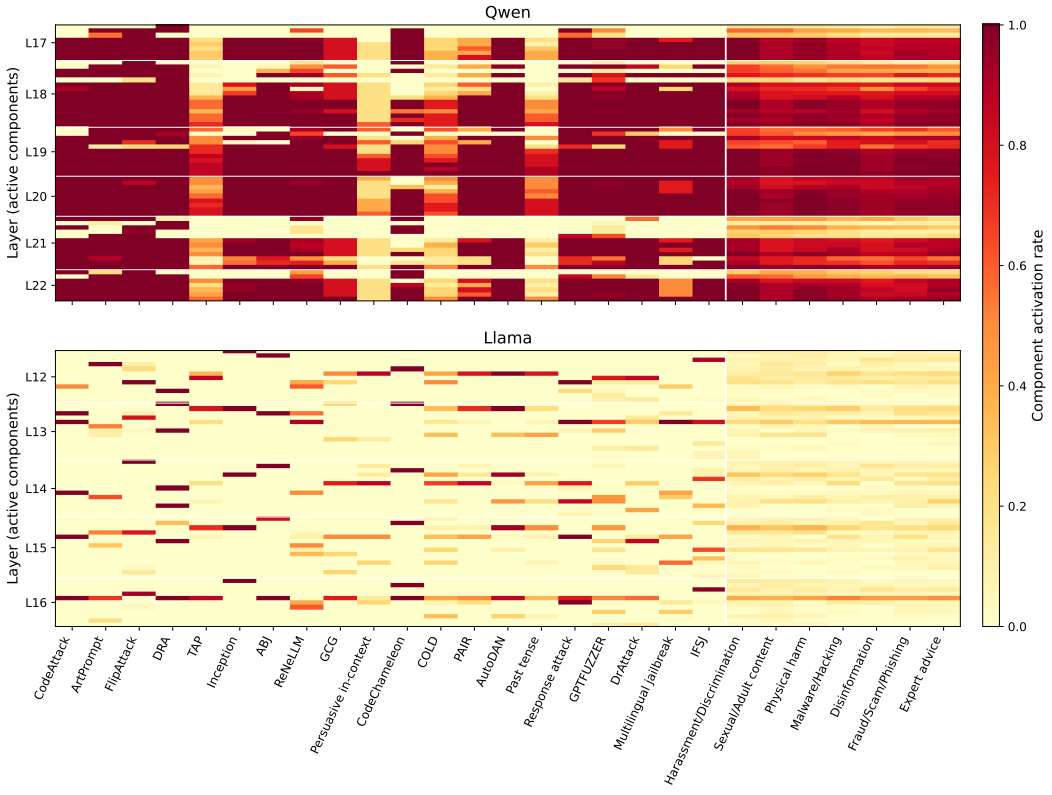


Figure 9: Fully labeled component-activation profiles corresponding to Figure 4(b). Columns show attack methods followed by harmful topics; rows show active layer–component pairs, grouped by layer. Colors give the fraction of inputs within each group that activate a component.

E COS METHOD AND ROUTING DIAGNOSTICS

E.1 ADDITIONAL DIAGNOSTICS FOR MAIN-TEXT FIGURES

Definitions for Figure 4. Panel (a) reports the fraction of all layer–component pairs whose pre-activation exceeds a given threshold on at least one harmful training example. Thresholds range from 0 to 0.4; the denominators are 384 pairs for Qwen and 320 for Llama. This measures use of the available component capacity, not attack selectivity. In panel (b), a layer–component pair is effective if its pre-activation exceeds the activation threshold on at least one harmful training example. Each heatmap cell gives the fraction of examples in the corresponding attack-method or harmful-topic group that activate the component; columns list attack methods first and topics second. Panel (c) averages the maximum component pre-activation at each layer within each label and normalizes each label's profile across layers to [0,1] (constant profiles are set to zero). Columns again list attack methods followed by topics; the two model-specific layer ranges are stacked vertically. The appendix component-sharing plot defines coverage as the fraction of harmful examples activating a steering component and compares its attack-method activation profile with those of other components.

E.2 LAYER SELECTION

Layer selection. Layer selection is based on harmful-versus-benign activation probes trained at each candidate layer. We compare probe performance across layers and select a contiguous high-performing layer band, where harmful and benign activations are most separable, for subsequent source steering vector learning, routing, and inference. The selected bands are layers 17–22 for Qwen3-8B and layers 12–16 for Llama-3.1-8B-Instruct. Figure 2(a) reports five-fold grouped-cross-validation AUROC for regularized linear probes on PromptSecurity-Eval prompt activations labeled by harmfulness (855 Qwen3-8B and 860 Llama-3.1-8B-Instruct training examples, grouped by underlying prompt identity). These harmful-versus-benign probes select intervention layers; they are distinct from the attack-method and harmful-topic probes used in the representation analysis.

E.3 ROUTER CALIBRATION

The intervention threshold is selected on calibration data to retain harmful coverage while suppressing benign interventions. Figure 2(c) divides each threshold by its selected value; the table below reports the absolute thresholds and intervention rates on calibration and held-out data. Router scores are model-specific and should not be interpreted as universal harmfulness probabilities.

Table 8: COS router thresholds and intervention rates (%). Higher harmful-input coverage (↑) and lower benign-input intervention (↓) are preferred; the threshold is a setting.

		Calibration		Held-out evaluation	
Model	Threshold	Harmful ↑	Benign ↓	Harmful ↑	Benign ↓
Qwen3-8B	0.145	99.43%	2.29%	96.00%	3.71%
Llama-3.1-8B-Instruct	0.105	100.00%	0.00%	99.44%	0.57%

E.4 COS HYPERPARAMETERS

Tables 9–11 report the COS settings used in the main defense experiments. The 700 benign training prompts are drawn from WildJailbreak.

Table 9: Source steering vector construction and training settings.

Setting	Qwen	Llama
Random seed	42	42
LLM precision	bfloat16	bfloat16
Intervention layers	17–22	12–16
Clustering layer	19	13
Clustering PCA dimensions	32	32
Maximum requests per cluster	5	5
Number of clusters	70	72
Source vector initialization	Zero	Zero
Optimizer	AdamW	AdamW
Learning rate	10^{-3}	10^{-3}
Weight decay	0.05	0.05
Batch size	4	4
Initial training epochs	20	20
Additional cluster-refinement epochs	$4 + 4 + 4$	4
Clusters refined in successive rounds	$4, 11, 4$	7
Refusal-prefix loss weight	1	1
Preference loss weight	0	0
Refusal-prefix training length limit	512 tokens	512 tokens
Refusal prefix	I'm sorry, but I can't assist with that request.	I can't help with that.

Table 10: Autoencoder (AE) and COS router training settings.

Setting	Qwen	Llama
AE dimensions (input / code / output)	$4096/64/4096$	$4096/64/4096$
Router dimensions (input / hidden / output)	$4096/512/64$	$4096/512/64$
AE code / router hidden / router output activation	ReLU / GELU / ReLU	ReLU / GELU / ReLU
Decoder bias	None	None
Optimizer	AdamW	AdamW
Optimizer (β_1, β_2); ϵ	$(0.9, 0.999)$; 10^{-8}	$(0.9, 0.999)$; 10^{-8}
Weight decay	10^{-4}	10^{-4}
Gradient clipping norm	1	1
AE pretraining epochs / learning rate	$200/10^{-3}$	$200/10^{-3}$
AE batch size (source vectors per layer)	70	72
Alternating optimization cycles	4	4
AE epochs per cycle / learning rate	$5/(5 \times 10^{-4})$	$5/(5 \times 10^{-4})$
Router epochs per cycle / learning rate	$2/(3 \times 10^{-4})$	$2/(3 \times 10^{-4})$
Router batch size	32	32
Decoder updated during router optimization	Yes	Yes
AE reconstruction MSE / cosine / norm weights	$1/0.5/1$	$1/0.5/1$
AE sparsity weight	0	0
Router code-matching weight	1	1
Router vector MSE / cosine / norm weights	$2/0.5/1$	$1/0.1/1$
Cluster classification weight / temperature	$0.3/0.07$	$0.3/0.07$
Benign code / vector penalty weights	$1/1$	$1/1$
Harmful activation margin / penalty weight	$0.12/40$	$0.12/40$
Benign activation margin / penalty weight	$0.05/30$	$0.05/30$
Harmful-feature Gaussian noise standard deviation	0.01	0.01
Noisy copies per harmful example	3	3
Feature normalization after noise	Unit L_2 norm	Unit L_2 norm
Auxiliary refusal-prefix loss weight	0.05	0.05
Auxiliary prefix batch size / length limit	$1/512$ tokens	$1/512$ tokens
Benign training prompts	700	700

Table 11: COS activation extraction, calibration, and inference settings.

Setting	Qwen	Llama
Router input pooling	Mean of final 8 prompt tokens	Mean of final 8 prompt tokens
Activation-extraction input length limit	2048 tokens	2048 tokens
Activation-extraction batch size	8	8
Calibration fraction	0.5	0.5
Calibration threshold grid	$[-0.05, 0.15]$, 81 points	$[-0.05, 0.15]$, 81 points
Calibration benign-rate target / harmful-rate minimum	$0.10/0.80$	$0.10/0.80$
Selected intervention threshold τ	0.145	0.105
Gate	Hard	Hard
Steering multiplier	1	1
Intervention position	Generated-token decoding	Generated-token decoding
Decoding	Greedy	Greedy
Maximum new tokens (PromptSecurity-Eval)	64	64

F Supplementary Behavioral Evaluations

F.1 Full XSTest Baseline Comparison

The main text compares Base and COS. This appendix gives the official string-match ORR for every baseline that appears in the PromptSecurity table. All values use the same 250 official XSTest safe prompts. CAST uses topic-conditioned steering; each baseline is evaluated with its model-specific configuration.

Table 12: Full XSTest baseline comparison. Cells report official string-match ORR (%, ↓: lower is better) on 250 safe prompts. The final column is the COS intervention rate; this gate-based intervention rate is reported only for COS.

Model	Base	CAA	BiPO	CAST	AdaSteer	COS ORR	COS intervention rate ↓
Qwen3-8B	4.00%	4.40%	2.40%	3.60%	0.00%	5.60%	3.20%
Llama-3.1-8B-Instruct	6.00%	5.20%	4.40%	0.00%	0.00%	6.00%	0.00%

Intervention sparsity complements refusal-based evaluation. The evaluated CAST configuration on Llama-3.1-8B-Instruct exhibits severe generation degeneration: despite a 0.00% string-match ORR, a stricter Qwen2.5 judge identifies a 93.60% safe-prompt failure rate, dominated by degenerate responses. Its low string-match ORR therefore does not establish preserved benign utility, since this metric misses failures without recognizable refusal phrases.

COS provides an additional, directly measurable form of preservation through its intervention gate. We report the intervention rate, the fraction of prompts for which the gate is activated, alongside ORR. On the 250 XSTest safe prompts, COS activates on only 3.20% of inputs for Qwen3-8B and 0.00% for Llama-3.1-8B-Instruct. Thus, 96.80% and 100.00% of these inputs, respectively, receive no COS activation modification: when the gate is inactive, the update in Equation 5 is exactly zero. This demonstrates sparse intervention on XSTest and makes preservation of the base model's computation explicit for those inputs. Intervention rate measures how often COS modifies activations, while ORR measures refusal behavior; the two metrics provide complementary views of benign-input behavior.

G More Related Work

Alignment with human values. Modern harmlessness work is built on large-scale instruction tuning and preference alignment, including RLHF, constitutional AI, and direct preference optimization (Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023).

Safety risks and jailbreak variation. Safety risks have expanded from direct harmful prompts to adversarial suffixes, persuasive attacks, disguise-and-reconstruction attacks, multi-step attacks,

code-format attacks, and ASCII-art attacks (Zou et al., 2023; Zeng et al., 2024; Deng et al., 2024; Liu et al., 2024c; Li et al., 2023a; Ren et al., 2024; Jiang et al., 2024). HarmfulQA, BeaverTails, JailbreakBench, and PromptSecurity-Eval provide different views of harmful content, jailbreak artifacts, and prompt-security evaluation settings (Bhardwaj & Poria, 2023; Ji et al., 2023; Chao et al., 2024; Hong et al., 2025).

Decoding-time and representation-level safety control. Several defenses try to improve safety without full model retraining. SafeDecoding increases harmless-token probability during early decoding (Xu et al., 2024b); Alignment-Enhanced Decoding refines token distributions by modeling the conflict between usefulness and harmlessness (Liu et al., 2024a); DRO learns trainable continuous safety prompts in hidden space (Zheng et al., 2024); and Circuit Breakers suppress representations responsible for harmful outputs (Zou et al., 2024).

Classical activation engineering. Activation engineering uses hidden-state interventions to control behavior with low training overhead. Representative methods include ActAdd, ITI, ICV, CAA, Mean-Centring, BiPO, and refusal-direction interventions (Turner et al., 2023; Li et al., 2023b; Liu et al., 2024b; Rimsky et al., 2024; Jorgensen et al., 2023; Cao et al., 2024; Arditi et al., 2024). They establish that compact directions can control model behavior, but most use fixed vectors or fixed intervention rules.

Adaptive activation engineering. Recent work increasingly adapts activation interventions across inputs, tokens, layers, or decoding states. CAST and AdaSteer condition refusal steering on input-side signals (Lee et al., 2025; Zhao et al., 2025); SADI constructs dynamic steering vectors from input semantics (Wang et al., 2025); DSEM retrieves token-level steering signals from episodic memory (Do et al., 2025); FairSteer applies dynamic steering to debiasing (Li et al., 2025); and newer controller-oriented papers study backtracking, layer selection, angular updates, ODE-style steering, feedback control, and dynamic rejection (Cheng et al., 2025; Sun et al., 2025; Vu & Nguyen, 2025; Zhao et al., 2026; Nguyen et al., 2025; Kang & Kim, 2026).

Autoencoder and feature-level representation analysis. Dictionary-learning and sparse-autoencoder work studies whether model representations can be decomposed into interpretable or reusable features (Yun et al., 2021; Lou et al., 2025; Kuznetsov et al., 2025; Xu et al., 2024a). PaCE uses autoencoder-derived concepts for safety control by identifying and suppressing harmful components (Luo et al., 2024). Other work studies when steering directions emerge and how interpretable steering features can be (She et al., 2025; Doan et al., 2026; Fedzechkina et al., 2026).

H Implementation of Threshold-Shift Prediction

We use layer-24 activations and a logistic safety score $s(h)$ trained on Original prompts, with feature scaling fitted on Original training prompts. For each of 32 method splits, eight attack methods are observed and six are held out. The 100 training and 100 test seed prompts are disjoint. The global offset g is the training mean of paired score differences $s(h_{\text{attack}}) - s(h_{\text{Original}})$. The predicted threshold is $g + r_\theta(h)$, combining the global offset with an input-dependent adjustment. Comparing $s(h)$ with this threshold is equivalent to evaluating the corrected score $s(h) - g - r_\theta(h)$ at zero. No attack-method identity or paired Original prompt is required at inference.

The linear predictor targets each observed method's mean paired score difference minus g. We center the method-averaged paired activation differences, remove their components along the top eight principal components fitted on Original training prompts, and retain their SVD span (at most seven dimensions). Projected features are standardized using attacked training examples. Ridge regression uses regularization 10, with each attacked sample weighted by $1/8$. Original-prompt anchors target $-g$ with weight 0.05 per sample, giving total Original-prompt loss weight 5% of the attacked loss. The nonlinear predictor targets individual paired score differences minus g. It uses 32 principal components fitted on attacked training activations, standardizes each component, and fits RBF kernel ridge regression with regularization 0.1 and kernel $\exp(-\|z - z'\|^2/m)$, where m is the median off-diagonal squared training distance. All transformations and targets are fitted using training data only.