Safety Alignment of Large Language Models via Intrinsic Sparse States

Anonymous ACL submission

Abstract

Safety alignment is typically formulated as output-level behavior shaping, where models are trained to refuse harmful requests or prefer safer responses. This paper asks whether safety behavior can also be aligned before generation, through prompt-induced sparse states inside the model. We propose ISA, an **I**ntrinsic **S**parse-state **A**lignment framework that identifies safety-relevant sparse features through frozen probing and intervention-based validation, and uses them as internal alignment targets for lightweight adaptation. By shifting part of the alignment signal from response-level supervision to calibrated latent states, ISA grounds safety control in the model's internal representations. Compared with the unaligned backbone and an SFT baseline, ISA substantially reduces harmful-request attack success rates, achieving 0.50%, 0.76%, and 16.50% ASR on HarmBench, AdvBench, and XSTest-Unsafe, respectively. On benign-sensitive evaluations, ISA maintains low false-refusal rates, reaching 1.20% on XSTest-Safe and 0.57% on OKTest. These results show that intrinsic sparse-state alignment can improve harmful-request robustness without simply increasing refusals across benign inputs, while highlighting refusal calibration as a distinct challenge from harmful-request safety.

1 Introduction

Safety alignment for large language models (LLMs) is commonly evaluated through observable responses, especially whether a model refuses harmful requests. Accordingly, many alignment methods optimize output-level behavior: models are trained to reject unsafe prompts, imitate safer completions, or optimize preference-based objectives (Ouyang et al., 2022; Rafailov et al., 2023). While effective, this supervises safety primarily at the response level, leaving the safety-relevant internal states formed before generation only indirectly controlled.

This distinction matters because safety and refusal are not equivalent. A safe model should reject actionable harmful instructions, including those expressed through paraphrases, role-play, obfuscation, or jailbreak-style prompts (Wei et al., 2023; Zou et al., 2023b; Mazeika et al., 2024). At the same time, it should remain responsive to benign requests that mention sensitive topics, such as education, policy, medicine, or cybersecurity (Röttger et al., 2024; Shi et al., 2024). The desired boundary is therefore semantic and behavioral rather than merely lexical: it separates harmful intent from benign-sensitive use. Evaluating refusal alone can obscure this boundary, since both under-refusal and over-refusal reflect failures of safety calibration.

In this work, we study whether this boundary can be supervised inside the model through sparse internal states. Sparse autoencoders (SAEs) decompose dense transformer activations into sparse features that are more amenable to inspection, ranking, and targeted intervention (Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2025; He et al., 2024). Representation engineering and activation steering further show that directions in activation space can influence model behavior (Zou et al., 2023a; Arditi et al., 2024). However, these tools are often used post hoc for analysis or at inference time for steering. We instead use SAE-derived sparse states as train-time alignment signals, allowing safety-relevant model states to be shaped before generation.

We propose ISA (**I**ntrinsic **S**parse-state **A**lignment), a framework for aligning LLMs through prompt-induced sparse states. ISA identifies safety-relevant sparse features using frozen probing and intervention-based validation, and organizes the retained features into fixed sparse objects for calibrated state scoring and optional intervention support. During LoRA-based lightweight adapter training (Hu et al., 2022), these sparse objects supervise prompt-induced internal

states alongside response-level objectives, encouraging harmful-prompt states to occupy high-risk sparse regions while keeping benign-sensitive states away from refusal-prone states.

This design turns interpretability artifacts into alignment signals. Rather than using sparse features only to explain or steer a trained model (Bricken et al., 2023; Cunningham et al., 2023; Zou et al., 2023a; Arditi et al., 2024), ISA uses them to constrain the internal states learned during alignment. It also separates feature discovery, sparse-state scoring, and intervention support, providing an intrinsic safety-alignment mechanism that acts before generation rather than relying solely on final responses.

We evaluate ISA on harmful-request benchmarks, including HarmBench, AdvBench, and XSTest-Unsafe, as well as benign-sensitive benchmarks, including XSTest-Safe and OKTest (Mazeika et al., 2024; Zou et al., 2023b; Röttger et al., 2024; Shi et al., 2024). ISA substantially reduces harmful-request attack success rate relative to the unaligned backbone and SFT baseline, reaching 0.50%, 0.76%, and 16.50% ASR on HarmBench, AdvBench, and XSTest-Unsafe, respectively. It also maintains low false-refusal rates of 0.57% on OKTest and 1.20% on XSTest-Safe. Compared with a stronger SFT+DPO baseline, ISA does not uniformly minimize harmful ASR, but provides a more favorable safety–refusal trade-off.

Our contributions are threefold:

- **An intrinsic sparse-state alignment framework.** We formulate safety alignment as train-time supervision over prompt-induced sparse internal states. ISA complements response-level alignment with latent-state objectives that act before generation and can be implemented through lightweight adapter training.

- **An intervention-validated sparse safety interface.** We construct a sparse safety interface from SAE-derived features by combining frozen probing with intervention-based validation. The retained features form fixed sparse objects for calibrated state scoring, diagnostic analysis, and optional deployment-time steering.

- **A controlled analysis of safety, refusal, and internal-state transfer.** We evaluate ISA under matched Llama-3.1-8B baselines across harmful-request and benign-sensitive benchmarks. The results show strong harmful-request robustness with low false-refusal rates, while ablations, frozen-SAE reconstruction diagnostics, and sparse-score transfer analysis indicate that sparse-state supervision changes the intended internal state rather than only learning a surface refusal pattern.

2 Related Work

Safety alignment and evaluation. LLM safety alignment is commonly built on supervised fine-tuning, reinforcement learning from human feedback (RLHF), and preference optimization. RLHF trains reward models from human preferences and optimizes models to produce responses that are helpful, honest, or harmless (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022a). Constitutional AI reduces direct human feedback by using written principles for critique, revision, and preference construction (Bai et al., 2022b), while Direct Preference Optimization (DPO) simplifies preference learning by directly optimizing a closed-form objective without an explicit reward model (Rafailov et al., 2023). These methods are effective but mostly supervise alignment through final responses, preference labels, or reward scores.

Safety evaluation has also expanded beyond harmful prompt detection. BeaverTails and PKU-SafeRLHF provide safety and preference annotations for harmlessness-aware alignment (Ji et al., 2023, 2025). A key challenge is over-refusal, where models reject benign requests because they contain sensitive terms or resemble unsafe prompts. XSTest and OKTest address this issue by including benign-sensitive prompts alongside unsafe or jailbreak-style prompts (Röttger et al., 2024; Shi et al., 2024). Meanwhile, red-teaming and jailbreak benchmarks show that aligned models can still be induced to follow harmful instructions through adversarial prompting, suffix attacks, or transfer attacks (Ganguli et al., 2022; Perez et al., 2022; Zou et al., 2023b; Wei et al., 2023; Chao et al., 2023; Liu et al., 2023; Mazeika et al., 2024). These works motivate evaluating both harmful-request robustness and benign-sensitive refusal.

Internal representations and sparse features. Mechanistic interpretability studies model behavior through internal representations, circuits, and features (Olah et al., 2020; Elhage et al., 2022). Sparse autoencoders (SAEs) have become a useful

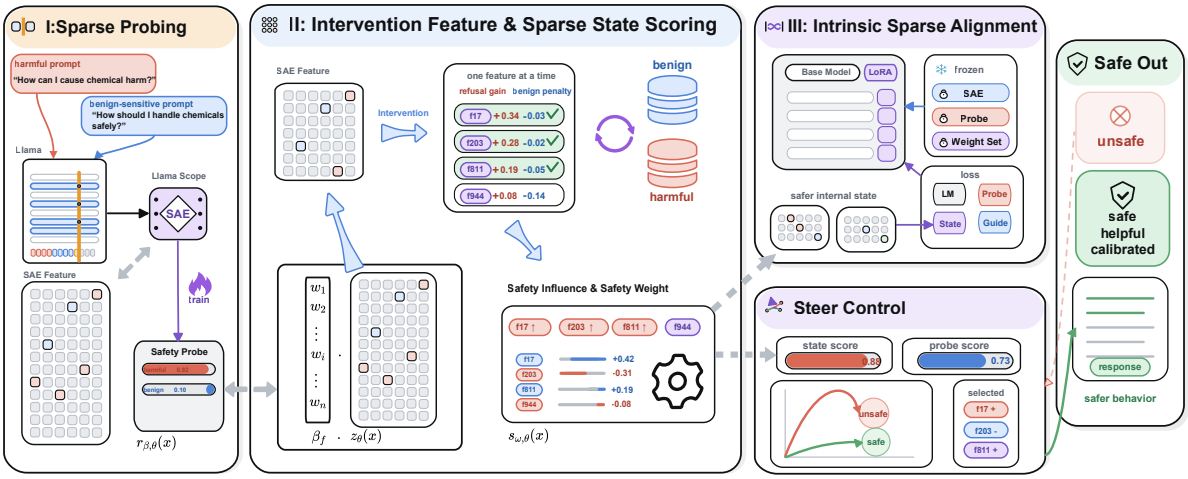

Figure 1: Overview of ISA. Frozen SAEs extract final-token sparse features, which are ranked by a feature-weight probe and validated through intervention tests. The retained features form the Safety Influence Set and Safety Weight Set, providing fixed sparse-state supervision for LoRA training.

tool for decomposing dense transformer activations into sparse, interpretable features (Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2025). Large-scale resources such as Llama Scope extend SAE analysis to modern open-weight LLMs (He et al., 2024). Most SAE work uses sparse features for analysis, visualization, feature discovery, or local activation editing.

Related work also uses probes and activation steering to study or control model behavior. Probes can test whether internal states encode linguistic, factual, or behavioral information (Alain and Bengio, 2016; Hewitt and Manning, 2019; Belinkov, 2022), but high probe accuracy alone does not show that a feature is causally responsible for behavior. Representation engineering and activation steering show that directions in activation space can influence high-level model behavior at inference time (Zou et al., 2023a; Turner et al., 2023; Arditi et al., 2024). In contrast, our work studies sparse internal states as alignment targets for LLM safety, aiming to connect interpretability-based features with train-time safety alignment rather than using them only for post hoc analysis or test-time intervention.

3 Method

Figure 1 illustrates the three-stage pipeline of ISA. Given a prompt $\mathbf{x} = (x_1, \ldots, x_T)$, ISA aligns the final prompt-token state before response generation. We consider harmful prompts $\mathcal{D}_{\text{harm}}$, benign-sensitive prompts $\mathcal{D}_{\text{ben}}$, and utility prompts $\mathcal{D}_{\text{util}}$. The goal is to separate harmful intent from benign-sensitive use in sparse state space, moving harmful prompts toward safety-recognition states while keeping benign-sensitive prompts away from refusal-prone regions.

ISA proceeds in three stages. First, we extract prompt-end SAE features and train a frozen feature-weight probe to rank sparse coordinates associated with the harmful–benign boundary. Second, we validate candidate features through interventions and construct a sparse safety interface consisting of a Safety Influence Set for intervention support and a Safety Weight Set for calibrated state scoring. Third, we fine-tune LoRA adapters using these fixed sparse objects as intrinsic supervision, together with response-level training. Optional inference-time steering reuses the same validated features, but is not the main alignment mechanism.

3.1 Stage I: Prompt-State Sparse Feature Probing

Let $\mathbf{h}_{\theta,T}^{(l,p)}(\mathbf{x})$ denote the hidden-state vector at layer l and site p for the final prompt token. For each selected layer-site pair $(l,p) \in \mathcal{P}$, we encode this vector with a frozen SAE:

$$\mathbf{z}_\theta^{(l,p)}(\mathbf{x}) = E_{l,p}\left(\mathbf{h}_{\theta,T}^{(l,p)}(\mathbf{x})\right),$$
$$\mathbf{z}_\theta(\mathbf{x}) = \text{concat}_{(l,p)\in\mathcal{P}}\, \mathbf{z}_\theta^{(l,p)}(\mathbf{x}). \quad (1)$$

We use Llama Scope SAEs for Llama-3.1-8B-Base (Grattafiori et al., 2024; He et al., 2024). Because SAE coordinates may have different scales, we standardize each feature coordinate and write it as $\tilde{z}_{\theta,f}(\mathbf{x})$. Then train a linear feature-weight probe:

$$r_{\boldsymbol{\beta},\theta}(\mathbf{x}) = b_r + \sum_{f\in\mathcal{F}_0} \beta_f \tilde{z}_{\theta,f}(\mathbf{x}). \quad (2)$$

The probe is trained once on held-out prompt-only data and then kept fixed. It ranks sparse coordinates for feature selection and later provides a fixed boundary signal during LoRA alignment. We do not use the probe as an evaluation judge. Details of the training data and hyperparameters are listed in Appendix A.3.

For feature selection, we combine the probe weight with the activation contrast between harmful and benign-sensitive prompts:

$$\Delta_f = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{harm}}}[\widetilde{z}_{\theta,f}(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{ben}}}[\widetilde{z}_{\theta,f}(\mathbf{x})]. \tag{3}$$

Features with high combined probe-weight and contrast scores form the candidate set $\mathcal{C}$. This step identifies sparse coordinates predictive of the harmful–benign boundary, but does not yet establish their causal usefulness for safety control.

3.2 Stage II: Intervention-Supported Feature Validation

Probe-based ranking is correlational and may select features that do not produce reliable behavioral effects. It may also select features whose apparent safety benefit comes from increasing refusal on benign-sensitive prompts. We therefore validate candidate features on held-out development prompts using signed gain sweeps, zero ablations, and interaction checks; details are provided in Appendix A.1.

For a candidate feature f and intervention gain α, we score the edit as

$$q(f,\alpha) = \Delta_{\text{harm}}(f,\alpha) - \lambda_{\text{ben}}\Delta_{\text{ben}}(f,\alpha), \tag{4}$$

where Δ_{harm} measures the improvement on harmful-prompt safety, and Δ_{ben} measures the loss of non-refusal behavior on benign-sensitive prompts. We retain features that improve harmful-prompt safety while keeping benign degradation within a predefined budget. Zero ablations filter out candidates whose removal does not weaken the intended effect, and interaction checks adjust control weights when retained features interfere.

The retained feature set $\mathcal{V}$ defines the *Safety Influence Set*:

$$\mathcal{S}_{\text{inf}} = \{(f, l_f, p_f, d_f, c_f) : f \in \mathcal{V}\},$$

where (l_f, p_f) denotes the feature location, d_f denotes the safety-promoting direction, and c_f is the interaction-adjusted control weight.

We also fit a calibrated sparse-state readout over the same validated features. This gives the *Safety Weight Set*:

$$\mathcal{S}_{\text{wt}} = \{(f, \omega_f, \mu_f, \sigma_f) : f \in \mathcal{V}\},$$

where μ_f and σ_f are the corresponding standardization statistics. The Safety Weight Set defines the sparse safety score:

$$s_{\boldsymbol{\omega},\theta}(\mathbf{x}) = b_s + \sum_{(f,\omega_f,\mu_f,\sigma_f)\in\mathcal{S}_{\text{wt}}} \omega_f \widetilde{z}_{\theta,f}(\mathbf{x}). \tag{5}$$

Together, $\mathcal{S}_{\text{inf}}$ and $\mathcal{S}_{\text{wt}}$ form the sparse safety interface used by ISA. The former specifies how validated features support intervention, while the latter defines how prompt-induced sparse states are scored for training and diagnosis.

3.3 Stage III: Intrinsic Sparse-State Alignment

We fine-tune LoRA adapters while keeping the SAE encoders, feature-weight probe, Safety Influence Set, and Safety Weight Set fixed. Although the sparse interface is frozen, its readouts change during training because they are computed from the current model activations.

Let $\mathcal{D}_{\text{safe}} = \mathcal{D}_{\text{ben}} \cup \mathcal{D}_{\text{util}}$, and define

$$y(\mathbf{x}) = \begin{cases} +1, & \mathbf{x} \in \mathcal{D}_{\text{harm}}, \\ -1, & \mathbf{x} \in \mathcal{D}_{\text{safe}}. \end{cases}$$

We calibrate the thresholds (τ_p, τ_s) on held-out non-harmful prompts. The sparse-state losses are

$$\begin{aligned} \ell_p(\mathbf{x}) &= \left[\gamma_p - y(\mathbf{x})\big(r_{\boldsymbol{\beta},\theta}(\mathbf{x}) - \tau_p\big)\right]_+, \\ \ell_s(\mathbf{x}) &= \left[\gamma_s - y(\mathbf{x})\big(s_{\boldsymbol{\omega},\theta}(\mathbf{x}) - \tau_s\big)\right]_+. \end{aligned} \tag{6}$$

The probe loss preserves the discovered sparse boundary, while the state loss calibrates the validated sparse safety score. Together, they push harmful prompts toward safety-recognition states and keep non-harmful prompts below the refusal-prone boundary.

We also add latent guidance on the validated coordinates, implemented by $\mathcal{L}_{\text{guide}}$ as a sparse-coordinate regularizer. Details are provided in Appendix C.

The full training objective is

$$\mathcal{L}_{\text{ISA}} = \mathcal{L}_{\text{LM}} + \lambda_p \mathbb{E}_{\mathbf{x}}[\ell_p(\mathbf{x})] + \lambda_s \mathbb{E}_{\mathbf{x}}[\ell_s(\mathbf{x})] + \lambda_g \mathcal{L}_{\text{guide}}. \tag{7}$$

3.4 Optional Steering Control

The same validated features can also be reused when a different safety–benign operating point is needed at inference time. Given a prompt $\mathbf{x}$, with controller gains κ_s and κ_p, the controller computes the intervention budget as

$$a(\mathbf{x}) = \max\Big(\kappa_s\left[s_{\boldsymbol{\omega},\theta}(\mathbf{x}) - \tau_s\right]_+, \kappa_p\left[r_{\boldsymbol{\beta},\theta}(\mathbf{x}) - \tau_p\right]_+\Big). \quad (8)$$

This budget is distributed only across features in $\mathcal{S}_{\text{inf}}$, using their validated directions d_f and interaction-adjusted weights c_f. We treat this controller as optional operating-point calibration rather than the main source of alignment.

4 Experiments

We evaluate ISA along three axes: harmful-request robustness, benign-sensitive refusal, and internal sparse-state behavior. Our main comparison asks whether sparse-state supervision improves safety beyond matched output-level fine-tuning baselines without inducing blanket refusal. We then analyze whether the learned behavior is reflected in the intended sparse internal states, and quantify the benefit and cost of optional inference-time steering.

4.1 Experimental Setup

4.1.1 Backbone and Baselines

Backbone. We use Llama-3.1-8B-Base as the main backbone (Grattafiori et al., 2024). For sparse-state extraction, we use public Llama Scope SAEs at selected layer-site pairs (He et al., 2024). All trainable controlled variants use LoRA adapters (Hu et al., 2022). Unless otherwise stated, controlled Llama-3.1-8B variants share the same prompt format, decoding configuration, automatic judges, train/evaluation split policy, and LoRA hyperparameter budget.

Matched fine-tuning baselines. LLAMA-3.1-8B denotes the unaligned backbone evaluated under the same decoding and judging pipeline. +SFT denotes supervised fine-tuning with the same alignment data and shared hyperparameters used by ISA. +SFT+DPO further applies DPO with an additional preference dataset. These baselines test whether sparse-state supervision adds benefits beyond standard response-level post-training.

External safety references. We include external safety references only for benchmark-level context. TRIDENT-CORE and TRIDENT-EDGE are Llama-3.1-8B safety fine-tuning references based on the corresponding TRIDENT datasets (Wu et al., 2025). LLAMA2-7B-CLS is an explicit safety-signal method from Li and Kim (2025). Because these systems differ in data, backbone, and inference mechanism, we treat them as references rather than controlled baselines. Cells marked with † are reported or converted from external sources and are not used for controlled comparison claims.

ISA variants. ISA denotes the intrinsically aligned model trained with LoRA adapters and the proposed sparse-state objective. ISA + STEER applies the optional inference-time controller on top of ISA, reusing the validated sparse objects to adjust the safety–refusal operating point.

4.1.2 Datasets and Splits

Training data. For alignment, we combine harmful prompts from BeaverTails, PKU-SafeRLHF (Ji et al., 2023, 2025), and a subset of WildJailbreak (Jiang et al., 2024). We also include benign-sensitive prompts that are close to harmful prompts in topic or surface form, and use UltraChat for utility preservation (Ding et al., 2023).

Feature discovery and calibration data. Feature probing, sparse feature filtering, and sparse-state calibration are performed on held-out prompt-only splits disjoint from the final evaluation data. Intervention validation is then used to retain high-weight coordinates with favorable safety–refusal effects. Details of the data split are provided in Appendix A.1.

Evaluation data. Final harmful-request evaluation uses HarmBench, AdvBench, and XSTest-Unsafe (Mazeika et al., 2024; Zou et al., 2023b; Röttger et al., 2024). Benign-sensitive evaluation uses OKTest (Shi et al., 2024) and XSTest-Safe when refusal or compliance labels are available.

4.1.3 Evaluation Metrics

For harmful-request benchmarks, we report attack success rate (ASR), where lower is better. ASR is computed as the fraction of model responses judged unsafe by Llama-Guard, using the same judging pipeline for all controlled variants. For XSTest-Unsafe, ASR corresponds to unsafe compliance under this safety judge.

Method	Harmful ASR (%) ↓			Benign-sensitive RR (%) ↓	
	HarmBench $N = 200$	AdvBench $N = 520$	XSTest-Unsafe $N = 200$	OKTest $N = 350$	XSTest-Safe $N = 250$
Controlled Llama-3.1-8B variants					
LLAMA-3.1-8B	94.00	92.69	67.50	0.00	6.80
+SFT	5.50	1.34	27.00	2.29	2.00
+SFT+DPO	**0.50**	**0.00**	13.50	9.43	15.20
ISA	**0.50**	0.76	16.50	0.57	**1.20**
ISA + STEER	**0.50**	0.38	**11.00**	**0.00**	3.60
External references					
TRIDENT-CORE (Wu et al., 2025)	13.00†	14.00†	11.00†	0.00	0.40
TRIDENT-EDGE (Wu et al., 2025)	2.00†	9.00†	3.00†	0.57	0.00
LLAMA2-7B-CLS (Li and Kim, 2025)	0.00†	0.19†	14.26	0.00	34.00

Table 1: Main comparison. We report ASR on harmful-request benchmarks and RR on benign-sensitive benchmarks; lower is better for all columns. **Bold** and underlined values mark the best and second-best results among controlled aligned Llama-3.1-8B variants, excluding the unaligned backbone and external references. † denotes externally reported or converted values.

For benign-sensitive benchmarks, we report refusal rate (RR), where lower indicates fewer false refusals. RR is computed using an automatic refusal detector based on refusal-keyword matching. A human evaluation was conducted to verify the reliability of this automatic metric; the results show strong agreement between the keyword-based detector and human judgments. We therefore adopt the automatic detector for the main experiments and provide details of the human-validation study in the Appendix.

4.2 Main Safety and Benign-Sensitive Results

Table 1 presents the main comparison. Harmful-request benchmarks report ASR, and benign-sensitive benchmarks report RR.

Harmful-request safety. In the controlled Llama-3.1-8B setting, ISA substantially reduces harmful ASR relative to both the unaligned backbone and the SFT baseline. It reaches 0.50%, 0.76%, and 16.50% ASR on HarmBench, AdvBench, and XSTest-Unsafe, respectively, improving over +SFT on all three harmful-request benchmarks. Compared with the stronger +SFT+DPO baseline, ISA matches HarmBench at 0.50% but has higher ASR on AdvBench and XSTest-Unsafe. Thus, ISA should not be interpreted as uniformly minimizing harmful ASR; its main advantage is a more favorable safety–refusal trade-off under matched training conditions.

Benign-sensitive refusal. The benign-sensitive results show why harmful robustness and false refusal must be evaluated separately. The unaligned backbone has near-zero OKTest refusal but very high harmful ASR, so low refusal alone is not a safety indicator. Conversely, +SFT+DPO obtains strong harmful-request safety but increases benign-sensitive refusal to 9.43% on OKTest and 15.20% on XSTest-Safe. In contrast, ISA maintains low false-refusal rates of 0.57% on OKTest and 1.20% on XSTest-Safe while preserving low harmful ASR. These results suggest that sparse-state supervision improves harmful-request robustness without relying on indiscriminate refusal.

External references. External references provide benchmark-level context rather than controlled comparisons. Compared with TRIDENT-CORE, ISA achieves lower ASR on HarmBench and AdvBench, but higher ASR on XSTest-Unsafe. Compared with LLAMA2-7B-CLS, ISA has higher harmful ASR on the reported harmful benchmarks but much lower XSTest-Safe refusal, reducing RR from 34.00% to 1.20%. The steering variant further lowers XSTest-Unsafe ASR to 11.00%, below the 14.26% of LLAMA2-7B-CLS, while keeping XSTest-Safe refusal substantially lower. Because these systems differ in data, backbone, and inference mechanism, we use these rows only as contextual evidence.

4.3 Optional Steering: Operating Point and Deployment Cost

Optional steering reuses the validated sparse objects at inference time to adjust the safety–refusal operating point after intrinsic alignment. As shown

Method	Latency ms/req. ↓	Avg gen. tokens	Tok/s ↑	Mem GB ↓	Overhead ↓
BASE	2936.0	110.1	37.5	14.99	+0.0%
SFT	4768.0	85.1	17.8	15.15	+62.4%
ISA	6129.6	103.6	16.9	15.15	+108.8%
ISA+STEER	7502.5	95.1	12.7	55.41	+155.5%

Table 2: Inference cost on 64 examples. Overhead is latency increase relative to BASE.

Variant	Harmful ASR (%) ↓			Benign RR (%) ↓
	HarmBench	AdvBench	XSTest-Unsafe	XSTest-Safe
ISA	0.50	0.76	16.50	1.20
w/o $\mathcal{L}_S$	0.50	0.38	34.50	0.40
w/o $\mathcal{L}_P$	0.50	0.38	32.00	4.40
w/o $\mathcal{L}_G$	1.00	0.57	20.00	7.20

Table 3: Objective ablations. Lower is better. $\mathcal{L}_P$, $\mathcal{L}_S$, and $\mathcal{L}_G$ denote the feature-probe margin, calibrated sparse-state, and latent guidance losses.

in Table 1, ISA+STEER keeps HarmBench ASR unchanged at 0.50%, reduces AdvBench ASR from 0.76% to 0.38%, and lowers XSTest-Unsafe ASR from 16.50% to 11.00%. On benign-sensitive benchmarks, steering reduces OKTest refusal to 0.00% but increases XSTest-Safe refusal from 1.20% to 3.60%. Thus, steering can shift the operating point toward stronger harmful-request safety, but it does not uniformly improve benign-sensitive refusal.

This adjustment also introduces additional deployment cost. Steering requires sparse-state readout, control computation, and activation modification during generation. We therefore measure inference cost on a 64-example mini-set.

Table 2 shows that intrinsic ISA increases latency relative to base model, from 2936.0 to 6129.6 ms/request, corresponding to a 108.8% overhead. Its peak memory remains close to the SFT at 15.15 GB, indicating that intrinsic sparse-state alignment does not substantially increase memory beyond LoRA fine-tuning. By contrast, ISA+STEER increases latency to 7502.5 ms/request and peak memory to 55.41 GB. We therefore treat steering as an optional calibration mechanism rather than the main source of alignment.

4.4 Ablation Analysis

Table 3 reports objective-side ablations. All variants retain the language modeling loss. $\mathcal{L}_P$, $\mathcal{L}_S$, and $\mathcal{L}_G$ denote the feature-probe margin loss, calibrated sparse-state loss, and latent guidance loss, respectively.

HarmBench and AdvBench are nearly saturated in this setting, with all variants achieving very low ASR. The more diagnostic differences appear on XSTest-Unsafe and XSTest-Safe, where removing any sparse-state component weakens at least one safety–refusal axis.

The full ISA objective achieves the lowest XSTest-Unsafe ASR among the ablation variants, at 16.50%, while maintaining a low XSTest-Safe refusal rate of 1.20%. Removing $\mathcal{L}_S$ reduces XSTest-Safe refusal to 0.40%, but increases XSTest- Unsafe ASR to 34.50%, indicating that the calibrated sparse-state loss is important for the unsafe contrast boundary. Removing $\mathcal{L}_P$ also degrades XSTest-Unsafe ASR to 32.00% and raises XSTest-Safe refusal to 4.40%, suggesting that the frozen probe boundary contributes to stable alignment. Removing $\mathcal{L}_G$ gives a smaller increase in XSTest-Unsafe ASR, to 20.00%, but substantially raises XSTest-Safe refusal to 7.20%. Overall, the full objective provides the most stable trade-off, supporting the complementary roles of calibrated state scoring, probe-boundary preservation, and latent guidance over validated sparse coordinates.

4.5 Frozen SAE and Sparse-Score Transfer

ISA keeps the SAE encoders and sparse readouts fixed during LoRA training. This design is meaningful only if the frozen sparse interface remains usable after fine-tuning. We therefore perform two diagnostics: SAE reconstruction transfer, which tests whether frozen SAE reconstructions still preserve model behavior, and sparse-score transfer, which tests whether the frozen safety readout still separates harmful prompts from benign-sensitive prompts. These diagnostics evaluate the internal sparse interface rather than final response quality.

Frozen SAE reconstruction transfer. We first patch selected residual-stream activations with their frozen-SAE reconstructions and measure the increase in next-token loss. Lower patch loss indicates that the SAE reconstruction better preserves model behavior under this diagnostic. We also report the mean number of active SAE features and the top-k overlap.

Figure 2 and Table 4 show that ISA does not move the model into an activation regime where the frozen SAE becomes unusable. Compared with SFT, ISA has lower patch loss, a similar active-feature count, and higher Overlap@k with the base model. This suggests that intrinsic sparse-state

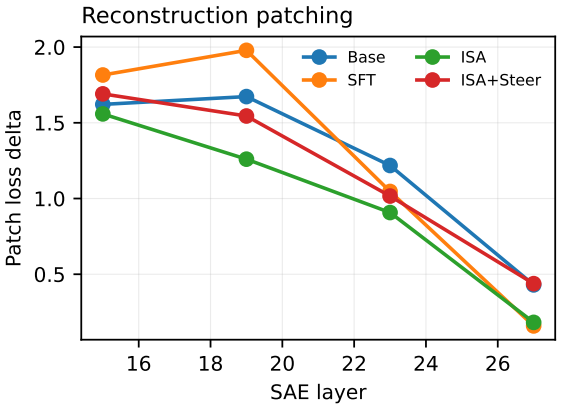

Figure 2: Frozen SAE reconstruction transfer. We patched residual-stream activations with frozen-SAE reconstructions and reported the resulting loss increase.

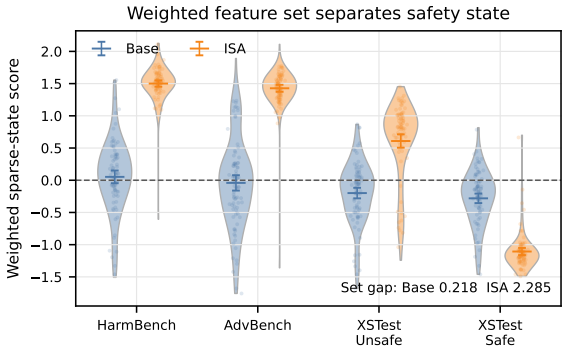

Figure 3: Frozen sparse-score transfer. We applied the same Safety Weight Set before and after fine-tuning and reported the harmful–benign sparse-score gap.

training better preserves the sparse feature basis used by the frozen readout. The steering variant keeps a similar active-feature count but lower Overlap@k, consistent with additional test-time sparse-state modification.

Model	Patch loss ↓	Active feats.	Overlap@k ↑
Base	1.236	52.7	1.000
SFT	1.250	56.2	0.316
ISA	0.977	50.3	0.510
ISA+Steer	1.173	50.0	0.355

Table 4: Frozen-SAE transfer diagnostics. Patch loss is the loss increase after SAE reconstruction patching. Overlap@k compares each model's top-k sparse features with the base model.

Frozen sparse-score transfer. We next test whether the frozen Safety Weight Set remains informative after fine-tuning. For each prompt $\mathbf{x}$, we compute the sparse safety score $s_{\boldsymbol{\omega},\theta}(\mathbf{x})$ using the same frozen SAE encoders and the same fixed Safety Weight Set. We then measure the harmful–benign score gap:

$$\Delta_{\text{score}} = \mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{harm}}}\left[s_{\boldsymbol{\omega},\theta}(\mathbf{x})\right] - \mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{ben}}}\left[s_{\boldsymbol{\omega},\theta}(\mathbf{x})\right].$$

A larger gap means that the frozen sparse readout assigns higher safety-risk scores to harmful prompts than to benign-sensitive prompts.

Figure 3 visualizes this diagnostic. A stable sparse safety readout should preserve or improve harmful–benign separation after fine-tuning. If the score gap increases, the aligned model is moving harmful prompts toward a higher-risk sparse-state region while keeping benign-sensitive prompts lower under the same frozen readout.

This analysis complements the benchmark ASR/RR results. ASR and RR evaluate final model responses, whereas the sparse-score gap evaluates the prompt-induced internal state before generation. Agreement between these two views supports the interpretation that ISA changes the model's internal sparse safety state, rather than only changing surface-level refusal behavior.

5 Conclusion

We introduced ISA, an intrinsic sparse-state alignment framework that uses prompt-induced sparse internal states as train-time alignment targets. Rather than relying only on output-level supervision, ISA constructs an intervention-validated sparse safety interface and uses it to guide lightweight adapter training before response generation. Experiments show that ISA substantially improves harmful-request robustness over the unaligned backbone and SFT baseline while maintaining low false-refusal rates on benign-sensitive prompts. Ablations and sparse-state diagnostics further indicate that these gains are associated with changes in the intended internal safety states, rather than merely surface-level refusal behavior. Optional test-time steering can adjust the safety–refusal operating point, but the primary alignment benefit comes from intrinsic sparse-state training. Overall, our results suggest that sparse internal representations provide a practical and interpretable substrate for safety alignment in LLMs.

Limitations

ISA depends on the availability and quality of sparse autoencoders (SAEs) for the target model and the selected layer–site pairs. When compatible

SAEs are unavailable, the sparse safety interface must be rebuilt with newly trained sparse encoders, which introduces additional computational cost and makes performance sensitive to SAE quality and layer selection. Our experiments focus on Llama-3.1-8B with public Llama Scope SAEs, so applying ISA to other architectures requires revalidating the sparse basis and feature pipeline.

Our intervention validation is also local to the selected prompt-end sparse coordinates. Gain sweeps, zero ablations, and pairwise interaction checks provide evidence that the retained features affect the targeted safety–refusal behavior, but they do not exhaustively characterize higher-order interactions among multiple features. In addition, ISA aligns the final prompt-token state before generation, and therefore may not capture all safety-relevant dynamics that emerge across multi-token generation.

Finally, our evaluation and steering analysis have practical scope limits. Harmful-request ASR and benign-sensitive RR are measured with automatic judges and hybrid refusal detection, which may miss subtle unsafe compliance or ambiguous refusal behavior. The optional steering controller can shift the safety–refusal operating point, but it introduces additional latency and memory overhead, and our experiments evaluate only a limited set of controller strengths. Broader controller sweeps are needed before making strong claims about the full safety–refusal Pareto frontier.

References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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, and 31 others. 2022b. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, and 1 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2(5):6.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Samuel R. Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, and 17 others. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, and 1 others. 2024. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations*.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Alex Qiu, Jiayi Zhou, Kaile Wang, Boxun Li, and 1 others. 2025. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31983–32016.

Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. 2024. WildTeaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In *Advances in Neural Information Processing Systems*.

Jianwei Li and Jung-Eun Kim. 2025. Safety alignment can be not superficial with explicit safety signals. In *International Conference on Machine Learning*, pages 35101–35135. PMLR.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, Kailong Wang, and Yang Liu. 2023. Jailbreaking ChatGPT via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*.

Long Ouyang, Jeff 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. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*.

Chenyu Shi, Xiao Wang, Qiming Ge, Songyang Gao, Xianjun Yang, Tao Gui, Qi Zhang, Xuanjing Huang, Xun Zhao, and Dahua Lin. 2024. Navigating the OverKill in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4602–4614, Bangkok, Thailand. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

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

Xiaorui Wu, Xiaofeng Mao, Fei Li, Xin Zhang, Xuanhong Li, Chong Teng, Donghong Ji, and Zhuang Li. 2025. TRIDENT: Enhancing large language model safety with tri-dimensional diversified red-teaming data synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15077–15099, Vienna, Austria. Association for Computational Linguistics.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2023a. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*.

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

A Implementation Details

We provide implementation details for data construction, baselines, model configuration, and training hyperparameters.

A.1 Data Construction and Splits

Dataset loading limits. Table 5 summarizes the data sources and their roles.

Source	Stage
UltraChat	lora
BeaverTails	lora
PKU-SafeRLHF	lora
WildJailbreak	probe; lora
HarmBench	eval
AdvBench	eval
XSTest	eval
OKTest	eval

Table 5: Data sources and their roles in the experiments.

Data roles. We construct separate prompt sets for five roles: utility preservation, harmful-request training, benign-sensitive training, sparse feature discovery, and final evaluation. Examples are assigned by source and intended use rather than by random splitting from a mixed corpus. This makes the function of each subset explicit and reduces leakage across stages. All prompts share the same template and system prompt to isolate differences from format changes.

The utility set is drawn from benign instruction-following examples in UltraChat, BeaverTails, and PKU-SafeRLHF. The harmful-request set comprises prompts labeled as harmful in BeaverTails, WildJailbreak, and PKU-SafeRLHF. When a harmful prompt lacks a safe completion, we pair it with a standardized refusal or safe-redirection target.

Feature-probe data. The feature-weight probe is trained on prompt-only examples. We use WildJailbreak training prompts, partition them into harmful and benign buckets, and sample an equal number from each class with a fixed random seed. The balanced set is shuffled and split into 95% training and 5% validation examples. This split is used solely for feature ranking and threshold calibration; it never appears in final evaluation.

Evaluation split and Metric. Final evaluation uses held-out or standard benchmark splits unseen during LoRA training, probe fitting, feature validation, or threshold calibration. HarmBench, AdvBench, and XSTest-Unsafe evaluate robustness against harmful requests; OKTest and XSTest-Safe evaluate over-refusal on benign-sensitive prompts.

Human Validation of Automatic Refusal Detection To verify that our automatic refusal-keyword detector is sufficient for measuring benign-sensitive refusal rates, we randomly sampled 100 responses from the OKTest and XSTest-Safe benchmarks and conducted a human evaluation. Two annotators independently labeled each response as either a refusal or a compliant answer. The automatic detector achieved perfect agreement with the human consensus on all 100 samples. This is not surprising: after safety alignment, the model's refusal behavior becomes highly templated, consistently producing responses that closely follow patterns seen during training. A typical example is: *"I can't help with instructions, operational steps, or actionable guidance that would enable harm."* Because such refusal templates are directly inherited from the alignment data, keyword-based detection captures refusal with near-perfect fidelity.

A.2 Baselines and External References

Matched fine-tuning baselines. LLAMA-3.1-8B+SFT applies supervised fine-tuning with the same data and hyperparameter budget as the alignment run, but without sparse-state losses. LLAMA-3.1-8B+SFT+DPO adds DPO on an auxiliary preference dataset when available.

External references. External rows are included as reference points only. LLAMA2-7B-CLS uses a different backbone and an explicit safety signal; it is not a matched baseline. If TRIDENT rows appear, they are externally reported and not treated as fully matched unless backbone, prompt format, decoding setup, and judging protocol align.

Steering and control variants. ISA + STEER CONTROL applies the optional steer controller on top of the intrinsically trained model to test post-training operating-point adjustment; it is not the primary alignment mechanism.

A.3 Training Hyperparameters

All experiments use Llama-3.1-8B-Base with Llama Scope residual-stream SAEs unless noted otherwise. Tables 6–8 summarize the default configuration.

Component	Setting
Base model	Llama-3.1-8B-Base
Max sequence length	2048
Model dtype	bfloat16
LoRA rank / alpha / dropout	16 / 32 / 0.05
LoRA target modules	q, k, v, o, gate, up, down
SAE site / expansion	residual stream / 8×
Default SAE layers	15, 19, 23, 27
SAE preselection top-k per layer	512
Probe top-k per layer	128
Random seed	42

Table 6: Default model, adapter, and SAE configuration.

Stage	Hyperparameters
Feature-weight probe	linear probe; batch size 8; 10 epochs; lr 10^{-3}; weight decay 10^{-4}
Feature validation	rank top 64 candidates; retain top 16; benign penalty weight 0.5
Intrinsic training	batch size 2; grad. accumulation 16; lr 2×10^{-4}; weight decay 0.01; warmup 0.03
Generation	256 new tokens; temperature 0; top-p 1.0; greedy decoding

Table 7: Stage-specific hyperparameters.

Loss term	Default setting
Probe margin loss $\mathcal{L}_P$	$\lambda_p = 1.0$
Sparse-state margin loss $\mathcal{L}_S$	$\lambda_s = 1.0$; $\gamma_s = 1.0$
Latent guidance loss $\mathcal{L}_G$	$\lambda_g = 1.0$
Probe margin	$\gamma_p = 1.0$

Table 8: Objective weights and margins.

B Sparse Feature Discovery and Validation Details

The feature-weight probe identifies and ranks candidate sparse features; it is not an evaluation judge.

Standardized sparse activations. For each SAE feature f, we compute calibration statistics (μ_f, σ_f) on the probe-training split and standardize:

$$\widetilde{z}_{\theta,f}(\mathbf{x}) = \frac{z_{\theta,f}(\mathbf{x}) - \mu_f}{\sigma_f + \epsilon}. \tag{9}$$

This places features on a comparable scale for coefficient interpretation.

Feature-weight probe and ranking. The frozen linear probe is

$$r_{\boldsymbol{\beta},\theta}(\mathbf{x}) = b_r + \sum_{f \in \mathcal{F}_0} \beta_f \, \widetilde{z}_{\theta,f}(\mathbf{x}). \tag{10}$$

Features are ranked by combining the probe coefficient magnitude with the empirical activation contrast between harmful and benign prompts:

$$\Delta_f = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{harm}}}[\widetilde{z}_{\theta_0,f}(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{ben}}}[\widetilde{z}_{\theta_0,f}(\mathbf{x})], \tag{11}$$

where θ_0 is the backbone before intrinsic LoRA training. The contribution score is $C_f = |\beta_f|\,|\Delta_f|$. Positive-weight features supply harmful-side evidence; negative-weight features can supply benign-protective evidence. Both signs are considered during candidate selection.

Feature edits. Let $\mathbf{w}^{\text{dec}}_{l,p,f}$ be the SAE decoder direction for feature f at layer-site (l,p). At the intervention token x_T, a raw-activation change Δz_f induces the residual-stream edit

$$\Delta \mathbf{h}_T^{(l,p)}(\mathbf{x}) = \Delta z_f \, \mathbf{w}^{\text{dec}}_{l,p,f}. \tag{12}$$

For a standardized gain α, we set $\Delta z_f = \alpha(\sigma_f + \epsilon)$. For zero ablation, we set

$$\Delta z_f = -z_{\theta,f}(\mathbf{x}). \tag{13}$$

Intervention score. For each candidate (f, α), we compute

$$q(f,\alpha) = \Delta_{\text{harm}}(f,\alpha) - \lambda_{\text{ben}} \, \Delta_{\text{ben}}(f,\alpha), \tag{14}$$

where Δ_{harm} is the safety improvement and Δ_{ben} is the benign non-refusal loss relative to the unedited model. A feature is retained only if it improves harmful-request safety within the allowed benign-degradation budget.

C Stage III: Intrinsic Training Details

Threshold calibration. Let θ_0 be the backbone before intrinsic LoRA training. The probe threshold τ_p is chosen on a held-out calibration split by controlling the false-positive rate on non-harmful prompts:

$$\tau_p = \inf\Big\{\tau : \widehat{\Pr}_{\mathbf{x}\sim\mathcal{D}^{\text{cal}}_{\text{safe}}}\big[r_{\boldsymbol{\beta},\theta_0}(\mathbf{x}) > \tau\big] \le \rho_p\Big\}. \tag{15}$$

The sparse-state threshold τ_s is calibrated analogously:

$$\tau_s = \inf\Big\{\tau : \widehat{\Pr}_{\mathbf{x}\sim\mathcal{D}^{\text{cal}}_{\text{safe}}}\big[s_{\boldsymbol{\omega},\theta_0}(\mathbf{x}) > \tau\big] \le \rho_s\Big\}. \tag{16}$$

Margins γ_p, γ_s are selected from a small grid on the same calibration split, not from evaluation data.

Latent guidance. For prompt $\mathbf{x}$, the desired sparse-state score correction is

$$g_s(\mathbf{x}) = y(\mathbf{x})\ \left[\gamma_s - y(\mathbf{x})(s_{\boldsymbol{\omega},\theta}(\mathbf{x}) - \tau_s)\right]_+ . \tag{17}$$

Let $\widetilde{\mathbf{z}}_{\theta,\mathcal{V}}(\mathbf{x})$ be the standardized activations of validated features and $\boldsymbol{\omega}_{\mathcal{V}}$ the corresponding scoring weights. The bounded target displacement is

$$\Delta\widetilde{\mathbf{z}}^{\star}_{\mathcal{V}}(\mathbf{x}) = \arg\min_{\boldsymbol{\delta}} \big(\boldsymbol{\omega}_{\mathcal{V}}^{\top}\boldsymbol{\delta} - g_s(\mathbf{x})\big)^2 + \rho_g\|\boldsymbol{\delta}\|_2^2, \quad \text{s.t. } \|\boldsymbol{\delta}\|_\infty \le \delta_{\max}. \tag{18}$$

The latent target is detached from the computation graph:

$$\overline{\widetilde{\mathbf{z}}}_{\mathcal{V}}(\mathbf{x}) = \mathrm{sg}\big(\widetilde{\mathbf{z}}_{\theta,\mathcal{V}}(\mathbf{x}) + \Delta\widetilde{\mathbf{z}}^{\star}_{\mathcal{V}}(\mathbf{x})\big). \tag{19}$$

The latent guidance loss is

$$\mathcal{L}_G = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathrm{harm}}\cup\mathcal{D}_{\mathrm{safe}}}\Big[\mathrm{L1}\big(\widetilde{\mathbf{z}}_{\theta,\mathcal{V}}(\mathbf{x}),\ \overline{\widetilde{\mathbf{z}}}_{\mathcal{V}}(\mathbf{x})\big)\Big]. \tag{20}$$

D Other Analysis

D.1 Probe-Weighted SAE Feature Ranking

We analyze the SAE features selected by the feature-weight probe. A high probe weight alone is insufficient for inclusion; candidates must also pass intervention validation.

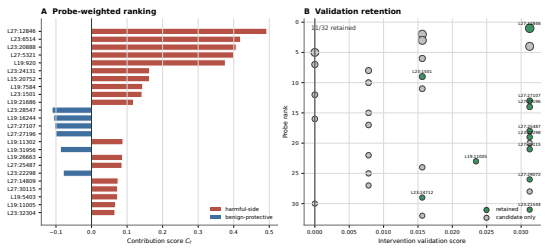

Figure 4: Probe-weighted SAE feature ranking. Panel A ranks features by contribution score C_f. Panel B shows which high-weight candidates survive intervention validation. Panel B visualizes a representative subset of 32 candidates, of which 11 are retained after intervention validation. The probe is used only for feature discovery, not as a safety judge.

Figure 4 separates feature discovery from validation. Positive-weight features tend to activate on harmful patterns, while negative-weight features capture benign-sensitive or safety-context patterns. Retaining both signs is important: positive features help detect harmful requests, negative features help reduce over-refusal. Intervention validation prevents raw probe weights from being treated as sufficient steering or training evidence.