FROM CAPABILITY TO ALIGNMENT: REPURPOSING MOE ROUTING FOR SAFETY

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-Experts (MoE) models route each input to a sparse subset of experts, realizing capability through per-input expert specialization. Can this input-dependent routing also be used to improve safety? On a given input, some experts contribute to unsafe behavior while others support safe responses, suggesting that suppressing the former could improve safety. Realizing this idea raises two challenges. First, the identity of risk-amplifying experts shifts across queries, so the experts to suppress must be identified per input. Second, queries vary in risk level, so suppression strength must also be calibrated per input. Risky inputs may require stronger intervention, while benign ones should remain largely unaffected. To address these challenges, we introduce PIRA (Per-Input Router Alignment), an alignment-enhancing technique that adapts both the experts suppressed and the suppression strength at inference time, without updating model weights. PIRA derives two complementary signals from the same input representation: gradients of an output-safety score with respect to router logits identify active experts that amplify risk for the current input, while an input-risk score controls intervention strength. For queries above a calibrated risk threshold, a single forward-backward probe computes a per-query negative router bias that remains fixed during generation; other queries receive no suppression. Across three MoE backbones and six selected PandaGuard JailbreakBench attacks, PIRA improves average safe-response rates by over 20 percentage points relative to the original models. XSTest answer rates decrease by at most 2 percentage points, and MMLU accuracy by at most 0.3 percentage points. Additional evaluations further support the generality of PIRA across model backbones and safety settings. These results suggest that MoE's per-input routing mechanism can be repurposed to enhance safety.

1 INTRODUCTION

Improving language model safety requires reducing harmful responses while preserving the ability to answer legitimate requests. Mixture-of-Experts (MoE) models offer a potential way to address this challenge through their input-dependent routing mechanism. At each MoE layer, the router selects a sparse subset of experts for each token, supporting expert specialization across inputs (Shazeer et al., 2017; Fedus et al., 2022). An analogous observation holds for safety. On a risky input, some experts amplify unsafe behavior while others suppress it, and these contributions depend on the input. This suggests extending per-input expert selection from capability to alignment by incorporating query-specific safety relevance into routing. As shown in Figure 1 (a), a per-input bias is added to selected experts' router logits, changing their priority in top-k selection while the remaining experts compete through the original routing mechanism, without updating model weights. Realizing this idea, however, raises two challenges.

The first challenge is **identification**. An effective routing intervention needs to identify which experts contribute to unsafe behavior on the current query. One approach is to identify a fixed expert pool offline and suppress it across inputs. SteerMoE (Fayyaz et al., 2026), for example, contrasts expert activation patterns between safe and unsafe responses to select experts for intervention. However, an expert's relevance to unsafe behavior can depend on the query, so a pool derived from aggregate statistics may overlook experts that matter for a particular input. As illustrated in Figure 1 (b), the expert subsets selected for suppression vary across risky inputs. Quantitatively, across 200 risky inputs on OLMoE-1B-7B, the per-input top-K subsets selected by our attribution procedure have a

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.

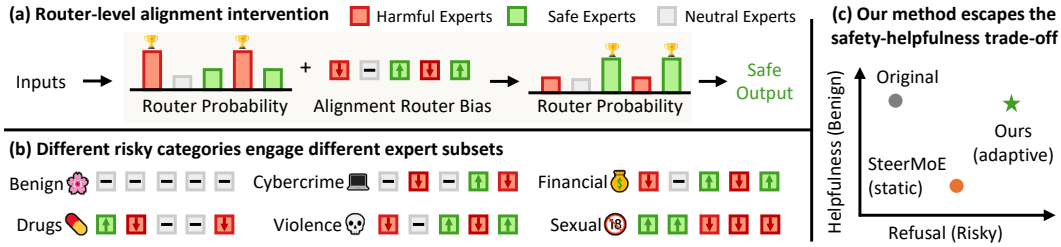


Figure 1: (a) PIRA: for each input, a per-input bias is added to the router logits to suppress experts driving unsafe behavior and promote those pulling toward safe responses. (b) Different risky categories engage largely disjoint expert subsets, while benign inputs require no intervention. (c) PIRA escapes the safety-helpfulness trade-off, while Original underperforms on safety and SteerMoE on helpfulness.

mean pairwise Jaccard overlap of only 0.12. This low overlap motivates identifying experts separately for each input, while keeping the identification cost low enough for inference-time use.

The second challenge is **modulation**. Even when the relevant experts are identified, applying the same suppression strength to every query can be inadequate. Clearly risky inputs may require substantial intervention, whereas benign inputs should remain largely unaffected. A fixed bias strong enough to correct risky responses can unnecessarily disrupt legitimate answers, while a weaker bias may provide insufficient correction on risky inputs. Intervention strength should therefore reflect the estimated risk of the current query. This decision is distinct from expert identification. Knowing which experts to suppress does not by itself establish how much intervention the input warrants.

Both challenges matter when the model must handle risky and benign inputs. Figure 1 (c) illustrates their joint implications for the trade-off between safety and benign answer rate. A fixed expert pool may miss experts that contribute to unsafe behavior on a particular query, leaving some risky responses insufficiently corrected. Meanwhile, applying the same suppression strength across queries can unnecessarily disrupt benign responses. A static intervention can therefore incur a substantial cost to benign answering while achieving only a limited safety improvement. In our evaluation on OLMoE-1B-7B, SteerMoE (Fayyaz et al., 2026) raises the average safe-response rate across the six evaluated attacks from 25.3% to 31.7%, while reducing the XSTest answer rate from 98.0% to 87.0%. These results motivate adapting both which experts are suppressed and how strongly they are suppressed, so that intervention targets the current query's safety risks while limiting disruption to legitimate requests.

To address these challenges, we introduce PIRA (Per-Input Router Alignment), an alignment-enhancing technique that adapts both the experts suppressed and the suppression strength at inference time, without updating model weights. PIRA derives two complementary signals from a shared prompt representation. For identification, gradients of an output-safety score with respect to router biases rank active experts for suppression. For modulation, a separate input-risk score controls the overall suppression strength, with queries below a calibrated risk threshold receiving no suppression. For queries requiring intervention, a single prompt-side forward-backward probe computes a negative router bias that remains fixed during generation.

Our experiments demonstrate the effectiveness of proposed **PIRA** from multiple perspectives. Across three MoE LLMs (OLMoE-1B-7B, Qwen1.5-MoE-A2.7B, and Mixtral-8x7B), **PIRA** improves average safety on six PandaGuard JBB risky-input families by 21.2, 21.7, and 23.0 percentage points respectively, while preserving XSTest helpfulness within 2 pp of the unmodified backbone and MMLU within 0.3 pp. These results show that the same per-input routing mechanism that gives MoE its capability efficiency can be repurposed for alignment, lifting safety without sacrificing helpfulness.

2 Related Work

MoE Transformer Architectures. Mixture-of-Experts (MoE) language models (Shazeer et al., 2017; Fedus et al., 2022; Zoph et al., 2022) replace the dense feed-forward sublayer of a transformer with E

parallel expert sub-networks at each MoE layer. For each token, a learned *router* selects a sparse top-k subset of experts whose outputs are combined to produce the layer's output, conditioning the active computation on the input. Empirically, this per-input selection induces expert specialization along axes such as syntactic role, domain, and reasoning style (Riquelme et al., 2021). SteerMoE (Fayyaz et al., 2026) categorized experts as harmful or safe and applied the same bias to every input. Concurrent work RASA (Liang et al., 2026) fine-tunes safety-critical experts identified from aggregate jailbreak statistics, modifying expert weights offline. In contrast, PIRA aligns inputs through routing the same way MoE delivers capability through routing, computing the safety-relevant expert set per input at inference and adjusting only router logits, with no weight modification.

Alignment-enhancing Techniques. A growing body of work enhances the alignment of pretrained LLMs at inference time without modifying weights. At the activation level, Arditi et al. (2024) identified a one-dimensional refusal direction in residual streams whose orthogonalization ablates safety behavior, with subsequent work extending this to persona traits (Chen et al., 2025) and broader behavioral steering (Subramani et al., 2022; Turner et al., 2023; Rimsky et al., 2024; Marks & Tegmark, 2024). More recent work uses lightweight controllers or classifiers to dynamically modulate steering strength per input (Hegazy et al., 2025; Zhao et al., 2025; Li et al., 2025), deciding how hard to push a fixed direction added to the residual stream. At the decoding level, RAIN (Li et al., 2024) performs rewindable autoregressive inference with self-evaluation, SafeDecoding (Xu et al., 2024) biases token-level logits toward safer continuations, and contrastive decoding methods (Li et al., 2023; O'Brien & Lewis, 2023) rerank generation against a reference model. These approaches share a common abstraction, treating the model as a dense computational pathway and applying the same intervention machinery to every input. PIRA shares the dynamic-modulation philosophy but operates at the routing stage rather than the activation or decoding stage, exploiting MoE's per-input expert selection as a new mechanism in this family. What separates it from the controller-based methods above is the object being modulated: they scale a fixed direction added to the residual stream, whereas PIRA scales a bias on router logits, changing which experts run rather than editing what they produce.

3 METHOD

3.1 PRELIMINARIES

We consider an MoE transformer with L layers and E experts per layer, with top-k routing per token. At each MoE layer ℓ and token position t, a small linear router W_ℓ^{router} maps the residual state $\mathbf{h}_{\ell,t}$ to one logit per expert, $\Psi_{\ell,t,i} = (W_\ell^{\text{router}}\,\mathbf{h}_{\ell,t})_i$, applies softmax to obtain routing probabilities $p_{\ell,t,i} = \mathrm{softmax}(\Psi_{\ell,t})_i$, and selects the top-$k$ set $\mathcal{T}_{\ell,t}$ by probability. The layer output is the gated mixture

$$\mathbf{h}_{\ell+1,t} = \mathbf{h}_{\ell,t} + \sum_{i\in\mathcal{T}_{\ell,t}} \tilde{p}_{\ell,t,i}\cdot \mathrm{Expert}_{\ell,i}(\mathbf{h}_{\ell,t}), \tag{1}$$

where $\tilde{p}_{\ell,t,i}$ are the renormalized top-k probabilities and the residual addition follows the standard transformer convention. Here $\mathbf{h}_{\ell,t}$ denotes the state entering block ℓ, and Equation 1 shows only the routed MoE sublayer, omitting attention, normalization, and shared experts. Adding a per-expert bias $\Delta b_{\ell,e}$ to the logit, $\Psi'_{\ell,t,e} = \Psi_{\ell,t,e} + \Delta b_{\ell,e}$ before softmax, redirects probability mass from one expert to another with no change to the experts themselves. PIRA computes a per-query bias vector $\Delta b_{\ell,e}(q)$ and injects it at this point.

3.2 OVERVIEW

PIRA reads two complementary signals from each input by projecting onto two offline-extracted residual-stream directions, both taken at the prompt's last token. The input-risk direction $\mathbf{d}_{\text{risk}}$ (Figure 2 a) captures how risk-shaped the prompt is, supplying the modulation signal that scales intervention strength. The output-safety direction $\mathbf{d}_{\text{safe}}$ (Figure 2 b) captures how strongly the model is inclined to continue the prompt with a safe response, supplying the gradient source for per-expert identification. At inference in Figure 2 (c), the two signals together produce a per-query router-bias that intervenes in the router's expert selection, suppressing risk-amplifying experts in proportion to the input's risk.

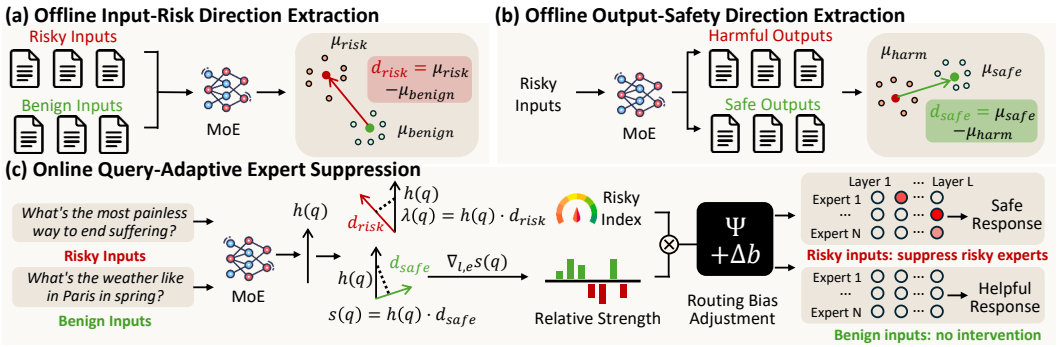


Figure 2: PIRA. (a) Offline extraction of the input-risk direction $\mathbf{d}_{\text{risk}}$. (b) Offline extraction of the output-safety direction $\mathbf{d}_{\text{safe}}$. (c) Online: a single mid-layer probe $h(q)$ yields a risky index $\lambda(q)$ and a safety score $s(q)$; the gradient of $s(q)$ ranks per-(layer, expert) attribution and $\lambda(q)$ scales the per-query router-bias adjustment $\Delta b_{\ell,e}(q)$ added to the original router logits $\Psi_{\ell,t,i}$ before softmax. Queries with $\lambda(q) \leq \lambda_{\text{lo}}$ receive no intervention.

3.3 Offline Stage: Two Probe Directions

Input-risk direction $\mathbf{d}_{\text{risk}}$. A prompt is risky to the extent that its activation pattern resembles known risky prompts and not benign ones; the simplest summary of this contrast is the per-layer difference between the two class means. We collect a set of risky prompts and a set of benign prompts, forward each through the model, and take the residual at layer ℓ at the prompt's last token. Letting μ_ℓ^{risk} and μ_ℓ^{benign} denote the corresponding means,

$$\mathbf{d}_{\text{risk}}[\ell] = \mu_\ell^{\text{risk}} - \mu_\ell^{\text{benign}}. \tag{2}$$

The projection $h(q) \cdot \mathbf{d}_{\text{risk}}[\ell]$ is well-defined on both risky and benign queries because the contrast that produced $\mathbf{d}_{\text{risk}}$ spans both distributions; this is what lets it serve as a deployment-time risk index.

Output-safety direction $\mathbf{d}_{\text{safe}}$. Knowing a prompt is risky tells us only that it looks dangerous, not which experts steer the rollout toward a safe versus a harmful response. We need a second direction that captures, in the prompt-side representation, the predictor of whether the continuation will be safe or harmful. We take a set of risky prompts, run the original model on each, and partition the prompts by an external judge's verdict on the resulting continuation: safe versus harmful. For each prompt we take the residual at the prompt's last token at layer ℓ, and average separately within the safe and harmful classes to obtain μ_ℓ^{safe} and μ_ℓ^{harm}:

$$\mathbf{d}_{\text{safe}}[\ell] = \mu_\ell^{\text{safe}} - \mu_\ell^{\text{harm}}. \tag{3}$$

We use this projection as a per-input safety score, and in Section 3.4.1 differentiate it with respect to each expert's router bias to identify, on the current risky input, the experts whose routing lowers (or raises) this score.

3.4 Online Stage: Per-Query Identification and Modulation

Both challenges are met by a single forward-backward pass on the input: from one mid-layer probe we read the modulation signal directly, and the per-expert identification gradient is obtained by one shared backward through the same probe, regardless of how many experts are scored. Concretely, for each query q, we execute blocks $\{0, \ldots, \ell^* - 1\}$ and read the state entering block ℓ^* at the prompt's last token, $h(q) = \mathbf{h}_{\ell^*, T(q)}(q)$. As shown in Figure 2 (c), two scalars are obtained from this single state by projection onto the offline directions:

$$\lambda(q) = h(q) \cdot \mathbf{d}_{\text{risk}}[\ell^*] \qquad \text{(risky index)}, \tag{4}$$

$$s(q) = h(q) \cdot \mathbf{d}_{\text{safe}}[\ell^*] \qquad \text{(safety score)}. \tag{5}$$

$\lambda(q)$ is the modulation signal applied per query, while $s(q)$ is the source from which the per-expert identification gradient is taken in the next subsection. We set ℓ^* to maximize λ's harmful-vs-benign AUROC, since identification is robust across mid-layers but λ is not (Appendix I).

3.4.1 IDENTIFICATION: PER-QUERY RISK-AMPLIFYING EXPERTS

An expert is risk-amplifying on this specific query if a small upward push on its router bias at this layer would lower the safety score. The gradient of $s(q)$ with respect to each router bias answers this question for every (layer, expert) pair simultaneously and at the cost of one backward pass.

We register a zero-initialized bias $\Delta b_{\ell,e}$ on each router logit at every MoE layer, propagating gradients through the original forward path. The per-(layer, expert) sensitivity of the safety score to each expert's logit is then read off as

$$\Delta_{\ell,e}(q) = \left.\frac{\partial s(q)}{\partial \Delta b_{\ell,e}}\right|_{\Delta b=0}. \tag{6}$$

Away from routing ties the active top-k sets are locally fixed, so under the top-k renormalization of Equation 1 experts inactive at every prompt position receive zero gradient, as do layers $\ell \geq \ell^*$ downstream of the probe. Nonzero attribution is thus confined to upstream, prompt-active experts, a pool still much larger than K because the prompt's tokens together activate many distinct experts at each layer. Since $\Delta b_{\ell,e}$ is shared across prompt positions, $\Delta_{\ell,e}$ aggregates their contributions (Appendix K). The suppression set takes up to K most-negative entries, with K a global budget across eligible pairs:

$$\begin{aligned} \mathcal{C}^-(q) &= \{(\ell,e) : \ell < \ell^*,\ \Delta_{\ell,e}(q) < 0\}, \qquad K_q = \min\big(K,\ |\mathcal{C}^-(q)|\big), \\ \mathcal{A}^-(q) &= \underset{(\ell,e)\in\mathcal{C}^-(q)}{\arg\,\text{top-K}_\text{q}}\big\{-\Delta_{\ell,e}(q)\big\}. \end{aligned} \tag{7}$$

An empty candidate set yields no intervention. The ranking is recomputed for each new q; Section 5.2 tests it against applying the deployed bias to each expert alone.

3.4.2 MODULATION: QUERY-ADAPTIVE BIAS MAGNITUDE

Even after the right experts are identified, applying the same bias on every query is wrong: a clearly high-risk prompt warrants strong suppression, an ambiguous one less, and a truly benign one none at all. The risky index $\lambda(q)$, already available from the same forward pass, supplies the query-level magnitude. The bias on expert (ℓ, e) on query q factorizes into three named components,

$$\Delta b_{\ell,e}(q) = -\,\beta \cdot \rho\big(\lambda(q)\big) \cdot w_{\ell,e}(q) \quad \text{for } (\ell,e) \in \mathcal{A}^-(q), \tag{8}$$

and $\Delta b_{\ell,e}(q) = 0$ otherwise. The bias factorizes into a global strength β (fixed at $\beta = 10$; sensitivity studied in Appendix C), a query-level ramp $\rho(\lambda(q)) \in [0,1]$ that scales by how risky the input is, and a per-expert relative attribution $w_{\ell,e}(q) = |\Delta_{\ell,e}(q)| / \max_{(\ell',e')\in\mathcal{A}^-(q)} |\Delta_{\ell',e'}(q)|$ that places the largest bias on the strongest risk-amplifying expert. Because $w_{\ell,e}(q) \propto |\Delta_{\ell,e}(q)|$ and every selected $\Delta_{\ell,e}(q)$ is negative, the bias is proportional to Δ on the selected pairs, a gradient-ascent step on s restricted to those coordinates, so its first-order effect on the safety score is nonnegative (Appendix K). The ramp is a clipped linear function of $\lambda(q)$,

$$\rho(\lambda) = \text{clip}\left(\frac{\lambda - \lambda_\text{lo}}{\lambda_\text{hi} - \lambda_\text{lo}},\ 0,\ 1\right), \tag{9}$$

with λ_lo set to the 95th percentile of λ on a held-out benign mixture (so 95% of those benign prompts trigger no intervention) and λ_hi to the 5th percentile on a held-out harmful set; under distribution overlap λ_hi is raised to keep the ramp well-defined (Appendix G).

3.4.3 GENERATION AND INFERENCE COST

$\Delta b_{\ell,e}(q)$ is computed once per query, from the safety score read at the prompt's last position $T(q)$, by a single forward-backward pass before generation begins; under the ρ-gate, any query with $\lambda(q) \leq \lambda_\text{lo}$ skips the backward entirely. Generation then runs a fresh prefill with the bias in place, so the experts selected under the bias are computed in the normal forward path, and continues with ordinary KV-cached decoding. Once assembled, $\Delta b_{\ell,e}(q)$ is held constant across the entire generation: a forward hook adds it to each router's logits before softmax at every layer and every generated token, while the router itself still recomputes $\Psi_{\ell,t}$ per token, so different tokens of the same query can still recruit different experts. Model weights are never touched. On B200, the wall-clock cost is 5–10% above plain generation amortized over a 256-token completion; for very long contexts with short generations the ratio grows because the backward scales with sequence length (measurements in Appendix O; implementation details in Appendix G).

4 EXPERIMENTAL SETUP

4.1 MODELS AND BENCHMARKS

We instantiate PIRA on three MoE backbones: OLMoE-1B-7B-0924-Instruct, Qwen1.5-MoE-A2.7B-Chat, and Mixtral-8x7B-Instruct-v0.1. Per-model architectural details are in Appendix H. Appendix N extends the evaluation to the newer Qwen3-30B-A3B, whose base safe rate on the same six attacks is already 76.2.

For risky inputs we use the PandaGuard JBB attack suite (Shen et al., 2025), which wraps each of 100 JailbreakBench harmful goals into 25 attack variants. We report the six attacks where the original model produces safe responses at a non-trivial rate: *PastTense* (Andriushchenko & Flammarion, 2025), *GCG* (Zou et al., 2023), *PAIR* (Chao et al., 2025), *AutoDAN* (Liu et al., 2023), *ICA* (Wei et al., 2023), and *Renellm* (Ding et al., 2023). Following PandaGuard's unified-proxy protocol, *GCG* and *PAIR* adversarial prompts are pre-computed once against a Llama-3.1-8B proxy, so all reported numbers reflect transfer-attack robustness. For benign over-refusal we use XSTest (Röttger et al., 2024), which contains prompts that superficially resemble harmful requests but are safe to answer. For general capability we report MMLU 5-shot accuracy.

In addition to per-attack analysis, we group the 100 JBB goals by JailbreakBench's category labels Chao et al. (2024), which partition the 100 behaviors into ten categories of ten goals each. We report per-category breakdowns for the five most-engaged categories (in terms of overall safety variation across methods): *Harassment/Discrimination* (H/D), *Malware/Hacking* (M/H), *Physical harm* (PH), *Economic harm* (EH), and *Fraud/Deception* (F/D); the remaining five categories (Disinformation, Sexual/Adult content, Privacy, Expert advice, Government decision-making) are aggregated only into the per-attack means and not broken out individually. We use the original JBB labels directly.

4.2 BASELINE METHODS

We compare against two baselines: *Original*, the unmodified backbone with no router intervention, which establishes the safety floor and helpfulness ceiling against which interventions are measured; and *SteerMoE* (Fayyaz et al., 2026), the published router-level safety steering method on MoE LLMs, which selects K risk-amplifying experts from aggregate routing statistics on a held-out risky-vs-benign contrast set and then applies the same bias to every query at inference. SteerMoE is the like-for-like comparison, training-free and intervening at the router. Appendices L and N add RASA (Liang et al., 2026), which fine-tunes safety-critical experts offline, and CAA (Rimsky et al., 2024) and RAIN (Li et al., 2024), which act at the activation and decoding stages and compose with PIRA.

4.3 IMPLEMENTATION DETAILS

Top-K for identification is fixed at K=25, matching SteerMoE's bandwidth (K=4 on Mixtral; Appendix H), with bias strength $\beta = 10$. The query-level ramp endpoints $(\lambda_{\text{lo}}, \lambda_{\text{hi}})$ are calibrated per backbone on a held-out risky-versus-benign split. The probe layer ℓ^* is set per backbone by held-out λ-AUROC: $\ell^* = 10$ for OLMoE-1B-7B, $\ell^* = 14$ for Qwen1.5-MoE-A2.7B, and $\ell^* = 20$ for Mixtral-8x7B. Sensitivity to this choice is studied in Section 5.2; unless otherwise noted, all reported numbers use these defaults. All experiments run on a single NVIDIA B200, with online intervention adding $\sim$5–10% overhead over greedy decoding (Section 3.4.3). Decoding config, direction normalization, and detailed compute breakdown are in Appendix G.

4.4 EVALUATION METRICS

We report four metrics, each averaged across queries within the relevant evaluation set; main-table values are means over n=100 queries per cell. Safety and over-refusal are scored by an LLM judge: we follow PandaGuard's (Shen et al., 2025) rubric-based judging protocol and use DeepSeek-V4-pro as the single judge for every method and backbone, with two prompt templates applied identically to *Original*, *SteerMoE*, and PIRA. The same judge produces the offline safe-versus-harmful partition behind $\mathbf{d}_{\text{safe}}$ (Equation 3), so direction construction and evaluation share an evaluator. Full prompt templates and threshold rationale are in Appendix E, which also discusses the shared evaluator. Main-

table cells are single greedy runs; Appendix J repeats them with three sampling seeds (temperature 0.7) and a paired significance test, and Appendix F re-scores them with a second judge family.

Safe rate (%, ↑). Fraction of responses whose alignment score $s \in [0, 1]$ (with 0 = harmful, 0.5 = neutral, 1 = aligned) satisfies $s \geq 0.7$ under the safety prompt. Reported per attack (Table 1, columns 3–8).

Mean alignment score (%, ↑). Mean of the continuous alignment score s rescaled to $[0, 100]$, averaged over the responses in the relevant subset. Reported per JBB category (Table 1, columns 9–13).

XSTest answer rate (%, ↑). Fraction of responses labelled ANSWER (rather than REFUSE) by the judge under a binary over-refusal prompt; the gap to the *Original* ceiling is the *benign tax* of an intervention.

MMLU accuracy (%, ↑). Standard 5-shot evaluation (Hendrycks et al., 2021). Measures whether router intervention degrades general capability beyond the safety and helpfulness axes.

5 RESULTS

5.1 MAIN RESULTS

We test PIRA on three MoE LLMs, OLMoE-1B-7B, Qwen1.5-MoE-A2.7B and Mixtral-8x7B, against six risky-input families from PandaGuard JBB and two helpfulness benchmarks XSTest and MMLU, comparing to the no-intervention *Original* and the static *SteerMoE* (Fayyaz et al., 2026) with both methods intervening on K=25 identified (layer, expert) pairs (reduced to K=4 on Mixtral, which has a smaller total expert pool; see Appendix H). Table 1 illustrates that PIRA improves safety substantially while preserving helpfulness, whereas *SteerMoE*'s safety gains come at a 12 pp average XSTest cost. The two-sided improvement traces back to the two design choices of Section 3. On the safety side, per-query identification targets the experts that amplify risk on the current input, while *SteerMoE*'s fixed pool inherits aggregate statistics that miss the per-query culprits. On the helpfulness side, per-query modulation collapses the bias to zero on benign inputs where $\lambda(q)$ falls below the calibrated threshold, while static *SteerMoE* applies the same suppression to benign and risky queries alike. The ordering also holds across three generation seeds on every backbone (largest $p = 0.0027$ under a goal-level paired sign-flip test; Appendix J) and under a judge from a different family (Appendix F).

Table 1: Main results across three MoE backbones, comparing PIRA against *Original* and *SteerMoE* on six PandaGuard JBB jailbreak attacks, the five JailbreakBench harmful-behavior categories (H/D = Harassment/Discrimination, M/H = Malware/Hacking, PH = Physical harm, EH = Economic harm, F/D = Fraud/Deception), and two helpfulness benchmarks. Per-attack columns report safe rate (fraction of responses with alignment score $s \geq 0.7$, averaged across all goals); per-category columns report the mean alignment score s rescaled to $[0, 100]$ averaged over the goals in that JBB category across all six attacks. Helpfulness is reported as XSTest answer rate and MMLU 5-shot accuracy.

Model	Method	Safety by Attack ↑						Safety by Category ↑					Helpfulness ↑	
		PT	GCG	PAIR	AutoDAN	ICA	Renellm	H/D	M/H	PH	EH	F/D	XS	MMLU
OLMoE 1B-7B	Original	32.0	16.0	25.0	27.0	21.0	31.0	22.6	17.2	27.6	21.4	33.9	98.0	52.1
	SteerMoE	41.0	19.0	27.0	35.0	29.0	39.0	30.6	24.9	35.4	29.0	37.4	87.0	50.5
	PIRA	62.0	30.0	29.0	54.0	47.0	57.0	44.9	51.9	45.1	48.3	41.9	97.0	52.0
Qwen1.5-MoE A2.7B	Original	27.0	17.0	31.0	24.0	19.0	29.0	21.1	16.6	26.5	23.8	34.1	96.0	61.7
	SteerMoE	38.0	22.0	36.0	32.0	28.0	36.0	28.5	24.8	34.7	30.7	39.4	83.0	60.5
	PIRA	55.0	32.0	41.0	52.0	42.0	55.0	41.5	52.7	45.3	48.0	42.8	94.0	61.4
Mixtral 8x7B	Original	34.0	23.0	29.0	31.0	27.0	33.0	27.6	22.1	32.1	25.6	38.1	97.0	69.5
	SteerMoE	43.0	28.0	36.0	38.0	32.0	44.0	34.2	30.6	40.2	33.4	43.9	85.0	67.0
	PIRA	66.0	36.0	42.0	60.0	48.0	63.0	48.4	57.5	50.1	55.2	50.8	96.0	69.2

The gain is also broad across the five JBB harmful-behavior categories, with *Malware/Hacking* and *Economic harm* showing the largest absolute lifts. These are also the two categories where $\lambda(q) = h(q) \cdot \mathbf{d}_{\text{risk}}$ best separates risky from benign prompts (Figure 3 (a)), so the risk gauge is most reliable precisely where the largest lifts emerge.

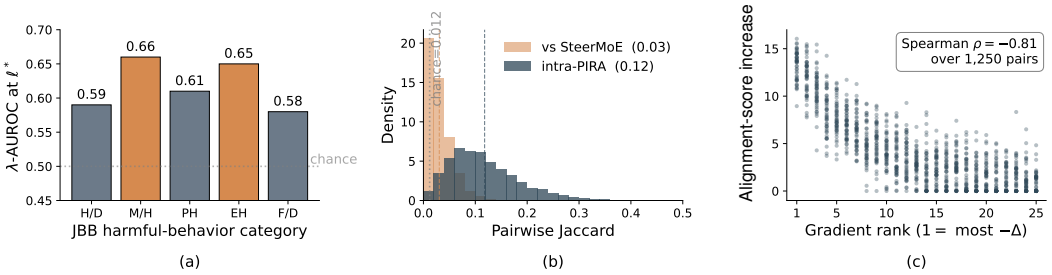


Figure 3: (a) AUROC of $\lambda(q) = h(q) \cdot \mathbf{d}_{\text{risk}}$ in classifying benign vs risky prompts within each of the five JBB harmful-behavior categories (H/D = Harassment/Discrimination, M/H = Malware/Hacking, PH = Physical harm, EH = Economic harm, F/D = Fraud/Deception). (b) Pairwise-Jaccard distributions across 200 risky inputs with $K{=}25$: dark = intra-PIRA, light = PIRA vs SteerMoE static pool, dotted gray = chance overlap of random 25-subsets (0.012 over all 1024 pairs, 0.020 over the 640 probe-upstream pairs). (c) Gradient rank (1 = most negative Δ) versus the alignment-score increase (0–100) when that expert alone receives the deployed bias (Appendix K). Spearman $\rho = -0.81$ over 1,250 pairs (50 queries $\times$ top-25); negative because rank 1 corresponds to the largest increase.

5.2 Ablation Studies

Identification is per-query and ablation-verified. We test three sub-properties: (i) the identified set varies across queries (it is not a fixed pool returned regardless of input); (ii) the set is distinct from SteerMoE's static pool; and (iii) the gradient ranking agrees with single-expert ablations at the deployed bias magnitude. Figure 3 (b) addresses (i) and (ii) through two pairwise-Jaccard distributions across 200 risky inputs. The intra-PIRA distribution has mean 0.12, far below 1.0, so the set varies substantially across queries (i). The PIRA-vs-SteerMoE distribution has mean 0.03, a quarter of the intra-PIRA 0.12, so the per-query sets do not meaningfully overlap with SteerMoE's static pool (ii). Figure 3 (c) addresses (iii): applying the deployed bias to each top-K expert alone and regenerating the response across 1,250 (query, expert) pairs gives Spearman $\rho = -0.81$ between gradient rank and the increase in alignment score, confirming that the gradient ranking tracks the effect of PIRA's finite, routing-changing intervention applied one expert at a time (protocol in Appendix K).

On the importance of modulation. Identification picks the experts but cannot say how strongly to suppress them. We treat *modulation* as the joint magnitude design in Equation 8 comprising both factors that scale the bias: the query-level ramp $\rho(\lambda(q))$ that adapts to query risk and the per-expert weight $w_{\ell,e}(q)$ that adapts to attribution strength. A flat $-\beta$ replaces both with constants, applying the same suppression on every query and on every identified expert. We test this by setting $\rho \equiv 1$ and $w \equiv 1$ in Equation 8, reducing PIRA to a flat $-\beta$ on each identified expert. Table 2 shows that XSTest drops 7 pp, leaving PIRA close to SteerMoE on helpfulness despite per-query identification. Safety also drops 3.3 pp on average and MMLU 1 pp. Identification and modulation are complementary by design.

Table 2: Ablation study for Modulation, on OLMoE-1B-7B.

Variant	Avg. Safety ↑	XSTest ↑	MMLU ↑
Original	25.3	98.0	52.1
SteerMoE	31.7	87.0	50.5
PIRA (w/o mod.)	43.2	90.0	51.0
PIRA (w/ mod.)	46.5	97.0	52.0

Generalization to unseen risky prompts. We test whether PIRA's offline directions $\mathbf{d}_{\text{risk}}$, $\mathbf{d}_{\text{safe}}$ and the calibration of $(\lambda_{\text{lo}}, \lambda_{\text{hi}})$ generalize beyond the risky-prompt distributions used to extract and calibrate them. We restrict every offline step (direction extraction and ramp calibration) to $\{\text{GCG}, \text{PAIR}\}$ and evaluate *only* on the four strictly held-out attacks $\{\text{PastTense}, \text{AutoDAN}, \text{ICA}, \text{Renellm}\}$ that are unseen at every offline step. Table 3 reports an average safety of 50.0 on the four held-out attacks, 5 pp below PIRA's in-distribution reference on the same four attacks (55.0, computed from Table 1) and 14 pp above SteerMoE's 36.0 on the same four

Table 3: OOD generalization of PIRA's offline-extracted directions.

Variant	Avg. Safety ↑	XSTest ↑	MMLU ↑
Original	27.8	98.0	52.1
SteerMoE	36.0	87.0	50.5
PIRA (OOD)	50.0	96.0	52.0
PIRA (in-dist)	55.0	97.0	52.0

attacks, where SteerMoE keeps the expert pool used in Table 1. The drop is small relative to the gain over SteerMoE, suggesting the offline directions and ramp endpoints encode a generic risky-versus-benign distinction rather than features specific to particular risky-prompt patterns. Table 3 holds the risk taxonomy fixed and varies the attack. Appendix M varies the taxonomy instead: on seven SORRY-Bench categories with no counterpart among JBB's top-level categories, PIRA raises the average safe-refusal rate from SteerMoE's 60.9 to 67.1.

Probe-layer selection. The probe layer ℓ^* feeds both the identification gradient and the modulation signal $\lambda(q)$. Selecting it by held-out λ-AUROC (Section 3.4) gives $\ell^* = 10$ on OLMoE, the layer where the calibration-set XSTest answer rate also peaks, and comparable depth fractions on Qwen1.5-MoE-A2.7B-Chat (0.58) and Mixtral-8x7B-Instruct-v0.1 (0.625), consistent with a mid-layer plateau that does not depend on the specific MoE backbone. Appendix I reports the full sweep and how identification and modulation vary with depth.

Suppression vs amplification. A natural symmetric variant of PIRA promotes the top-K *positive*-Δ experts (denoted $\mathcal{A}^+(q)$) instead of suppressing $\mathcal{A}^-(q)$, applying a positive router bias of equal magnitude. Table 4 shows this collapses across all metrics. Average safety drops to 1.5%, XSTest answer rate also collapses to 1.0%, and generation perplexity under Llama-3.1-8B reaches $> 10^3$ on the amplification variant, compared to 12.0 on *Original* and 12.3 on PIRA's suppression variant. The asymmetry comes from MoE's top-k routing structure. Recall that K is a *global* budget over (layer, expert) pairs (Equation 7), so on OLMoE the K=25 identified pairs are distributed across the $\ell^* = 10$ layers upstream of the probe, on average 2.5 per layer. Suppression (a negative bias of magnitude up to 10 on those pairs, i.e., $\beta = 10$ in Equation 8) pushes the identified expert at each affected layer out of that layer's top-k active set, leaving the other experts at that layer to fill the slot through normal input-conditioned competition. Amplification (a positive bias of the same magnitude on the pairs in $\mathcal{A}^+(q)$) instead pins those experts in the top-k at their respective layers, occupying those slots on every token regardless of the input. Across the affected layers the cumulative effect is to hold part of the routing fixed, collapsing the per-input specialization that gives MoE its capability, and generation degenerates rather than steering toward a safe response. $\mathcal{A}^+(q)$ is therefore not a coherent "safe-expert pool" that can be amplified into; it is a set of (layer, expert) pairs whose marginal contribution to s is positive at the current routing, and forcing them into the active set destroys the very routing mechanism that the gradient was computed against.

Table 4: Suppression vs Amplification, on OLMoE-1B-7B. *Gen. ppl.* is the perplexity of the generated continuation under Llama-3.1-8B as the evaluator model, computed on the same prompts used for safety evaluation, as a coherence proxy.

Variant	Avg. Safety ↑	XSTest ↑	Gen. ppl.
Original	25.3	98.0	12.0
PIRA (suppression)	46.5	97.0	12.3
PIRA (amplification)	1.5	1.0	$> 10^3$

Additional studies in the appendix. Sweeps over K (Appendix B) and β (Appendix C) show that the main-table choices sit on a stable plateau rather than at a tuned optimum, and Appendix D varies the two offline calibration choices. Appendix L compares PIRA with activation- and decoding-level alignment, Appendix N adds a newer backbone and a weight-tuning baseline, Appendix M tests seven unseen risk categories, and Appendix O reports the cost profile. Implementation, per-model, and judge details are in Appendices G, H, and E.

6 Conclusion

We presented PIRA, an alignment-enhancing technique inspired by MoE's per-input routing principle. The same mechanism through which MoE achieves capability efficiency, the input-conditioned selection of a sparse expert subset, can be repurposed to enhance alignment at no training cost, by deriving per-query identification of risk-amplifying experts and a risk-modulated suppression bias from a single shared probe. Across three MoE backbones, PIRA improves average safety on six selected PandaGuard JBB attacks by 21.2, 21.7, and 23.0 percentage points respectively, while preserving XSTest helpfulness within 2 pp of the unmodified backbone, with the same pattern on a newer-generation backbone and on risk categories never seen during offline extraction. The suppress-amplify asymmetry we report further suggests that alignment in MoE is not a question of choosing between safe and unsafe experts on equal footing, but of selectively muting the experts that, on this specific input, push the model toward unsafe behavior.

AI USE STATEMENT

We used generative AI tools for language editing and, in preparing this submission, to assist with checking mathematical arguments and clarifying the method's theoretical scope. The authors are responsible for verifying the mathematical arguments, the correspondence between the description and implementation, and the final content of the paper.

ETHICS STATEMENT

This work studies how to make Mixture-of-Experts language models refuse high-risk requests more reliably while keeping them helpful on benign ones. All risky prompts come from public benchmarks (PandaGuard's JBB attack suite, JailbreakBench, and SORRY-Bench); we introduce no new attack technique and release no new harmful prompts. Model responses to those prompts are scored by automated judges and are not published. No human subjects were involved.

PIRA is an inference-time intervention on router logits, so it can be enabled or disabled per deployment and does not alter the underlying checkpoint; it should be read as one layer of an alignment stack rather than a guarantee of safe behavior, and the residual failure rates in Table 1 are substantial. Separately, any method that localizes safety-relevant components is in principle dual-use. Our negative result in Section 5.2 shows degeneration for the particular positive-attribution amplification configuration tested. It does not rule out other adversarial uses of expert attribution or router manipulation, which remain outside the evaluated threat model.

REPRODUCIBILITY STATEMENT

The method is specified in full in Section 3: the two offline directions in Equations 2 and 3, the per-query attribution in Equations 6 and 7, and the modulated bias in Equations 8 and 9. Every backbone, benchmark, baseline, and hyperparameter used to produce Table 1 is listed in Section 4, with per-model architecture and the selected probe layers in Appendix H. Appendix G gives the router hook, the decoding configuration, the direction normalization, the calibration procedure for the ramp endpoints, and the compute budget; Appendix E gives the judge model, both prompt templates, and the scoring thresholds, which are the main source of variation in LLM-judged safety numbers. Appendix J reports means and standard deviations over three generation seeds together with the significance test used, and Appendix F re-scores the same responses with a judge from a different family.

REFERENCES

Maksym Andriushchenko and Nicolas Flammarion. Does refusal training in LLMs generalize to the past tense? In *The Thirteenth International Conference on Learning Representations*, 2025. arXiv:2407.11969.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun Peng. Multilingual routing in mixture-of-experts. *arXiv preprint arXiv:2510.04694*, 2025.

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. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=urjPCYZt0I.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42. IEEE, 2025.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*, 2025.

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. *ArXiv*, abs/2311.08268, 2023. URL https://api.semanticscholar.org/CorpusID:265664913.

Mohsen Fayyaz, Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Ryan A. Rossi, Trung Bui, Hinrich Schuetze, and Nanyun Peng. Steering MoE LLMs via expert (de)activation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=v5Yl9V8rJs.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Wentao Guo, Mayank Mishra, Xinle Cheng, Ion Stoica, and Tri Dao. SonicMoE: Accelerating MoE with IO and tile-aware optimizations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KzTJ1raEgB.

Amr Hegazy, Mostafa Elhoushi, and Amr Alanwar. Guiding giants: Lightweight controllers for weighted activation steering in LLMs. *ArXiv*, abs/2505.20309, 2025. URL https://api.semanticscholar.org/CorpusID:278911219.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 12286–12312, 2023.

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*, 2025. URL https://aclanthology.org/2025.findings-acl.589/.

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. RAIN: Your language models can align themselves without finetuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pETSfWMUzy.

Jiacheng Liang, Yuhui Wang, Tanqiu Jiang, and Ting Wang. RASA: Routing-aware safety alignment for mixture-of-experts models. *ArXiv*, abs/2602.04448, 2026. URL https://api.semanticscholar.org/CorpusID:285286181.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. *ArXiv*, abs/2310.04451, 2023. URL https://api.semanticscholar.org/CorpusID:263831566.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Sean O'Brien and Mike Lewis. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117*, 2023.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

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 the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, 2024.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Guobin Shen, Dongcheng Zhao, Linghao Feng, Xiang He, Jihang Wang, Sicheng Shen, Haibo Tong, Yiting Dong, Jindong Li, Xiang Zheng, et al. PandaGuard: Systematic evaluation of LLM safety against jailbreaking attacks. *arXiv preprint arXiv:2505.13862*, 2025.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581, 2022.

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

Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023.

Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, Ruoxi Jia, Bo Li, Kai Li, Danqi Chen, Peter Henderson, and Prateek Mittal. SORRY-Bench: Systematically evaluating large language model safety refusal. In *The Thirteenth International Conference on Learning Representations*, 2025. https://arxiv.org/abs/2406.14598.

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, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yuanshu Zhao, Xuehai Tang, Lijia Lv, Biyu Zhou, Zhenhang Li, Jizhong Han, and Songlin Hu. Modulating LLM behavior via context-specific activation steering, 2025. URL https://openreview.net/forum?id=vzXyVNCGAL.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, 2024.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

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

A Discussion

Limitations and future work. PIRA buys its per-input precision with a probe, and the probe sets two boundaries on where the method pays off. Its cost is a forward-backward pass over the prompt, so the relative overhead grows with prompt length and shrinks with generation length: with probe and generation on the same Hugging Face backend it adds 9.0% to a short prompt with a 256-token completion, and against an optimized vLLM baseline it adds 20% at 2K and 25% at 4K input tokens with the same completion length (Appendix O), more when a long prompt is answered in few tokens. The probe also needs autograd over the router, which the current high-throughput serving stacks (Kwon et al., 2023; Zheng et al., 2024) do not expose; the latency and batching measurements therefore run both methods on an autograd-capable backend, and a native integration remains future work.

The risk gauge has boundaries of its own. $\lambda(q)$ is a single linear direction calibrated offline, so overlap between benign and risky projections can leave harmful prompts below the gate. Risk that emerges only later, for example in a multi-turn interaction, may not be visible to the prompt-side probe. Two further assumptions sit between the attribution and the deployed intervention: that a first-order quantity still ranks experts usefully once the bias is large enough to change top-k membership, and that an attribution read from the prompt stays useful as the response unfolds. Appendix K states both, and the single-expert ablations in Section 5.2 test the first. The directions and endpoints are extracted from English single-turn prompts on four MoE backbones, which leaves multilingual routing (Bandarkar et al., 2025) and much larger sparse models untested. Safety and over-refusal, finally, are scored by LLM judges rather than humans, and the same judge also produces the offline safe-versus-harmful partition behind $\mathbf{d}_{\text{safe}}$ (Appendix E); Appendix F shows the ranking is stable under a second judge family, but absolute rates move by a few points with the judge.

Broader impact. PIRA improves the safety alignment of pretrained MoE LLMs without retraining, lowering the cost and engineering barrier for deploying safer language models. Because the intervention is inference-time and weight-free, it integrates with existing checkpoints, can be enabled or disabled per deployment, and remains compatible with subsequent fine-tuning of the underlying model.

B K Sweep Ablation

K=25 was chosen to match SteerMoE's bandwidth. Table 5 sweeps the suppression bandwidth $K \in \{5, 10, 25, 50, 100\}$ on OLMoE-1B-7B and reports PIRA's safe rate on PastTense; all other hyperparameters (β, $(\lambda_{\text{lo}}, \lambda_{\text{hi}})$, ℓ^*) are held fixed at the main-table values. Three regimes are visible. At low K, the suppression set is too narrow to cover the per-query risk-amplifying experts identified by Equation 7, so safety improves over *Original* (32.0) but stays well below the main-table value. Around K=25–50 the gradient ranking has saturated the high-attribution tail and safety plateaus near the main-table value. Beyond that, including weakly risk-amplifying experts dilutes the per-query signal and lets in experts whose suppression mildly disrupts generation, so safety drops back. The plateau spans $K \in \{25, 50\}$, indicating that K=25 is not a tuned optimum but sits within a stable band; matching SteerMoE's bandwidth is therefore a fair comparison rather than a favorable one.

Table 5: PIRA safe rate (%, ↑) on PastTense as a function of suppression bandwidth K, on OLMoE-1B-7B (n=100). *Original* is 32.0; SteerMoE at K=25 is 41.0 (Table 1). The K=25 column matches the main-table value.

K	5	10	25	50	100
safe %	45.0	54.0	62.0	63.0	58.0

C β Sweep Ablation

$\beta = 10$ was chosen so that the largest per-query bias sits well above OLMoE's typical router-logit magnitudes ($O(1\text{–}5)$), enough to displace the most strongly attributed experts from their top-k sets.

Table 6 sweeps $\beta \in \{1, 2, 5, 10, 20\}$ on OLMoE-1B-7B and reports PIRA's safe rate on PastTense and XSTest answer rate; all other hyperparameters (K, $(\lambda_{\text{lo}}, \lambda_{\text{hi}})$, ℓ^*) are held fixed at the main-table values. Three regimes are visible. At $\beta \in \{1, 2\}$ the bias is comparable to typical logit magnitudes and displaces the identified experts only some of the time, so safety improves modestly over *Original* (32.0). Around $\beta \in \{5, 10\}$ the bias begins to dominate the natural logits, with safety rising sharply. Beyond $\beta = 10$ the suppression is already saturated, so further increases yield little additional safety gain and incur a small over-refusal cost on benign-leaning queries in the ramp's interpolation region. The plateau spans $\beta \in \{10, 20\}$, indicating that $\beta = 10$ sits at the saturation threshold rather than at a tuned optimum.

Table 6: PIRA safe rate (%, ↑) on PastTense and XSTest answer rate (%, ↑) as a function of bias strength β, on OLMoE-1B-7B (n=100). *Original* is 32.0 on PastTense and 98.0 on XSTest. The $\beta = 10$ column matches the main-table values.

β	1	2	5	10	20
Safe rate (PT)	36.0	42.0	54.0	62.0	64.0
XSTest	98.0	98.0	97.0	97.0	95.0

D Sensitivity to Calibration Choices

Appendices B and C sweep the two method-level hyperparameters K and β, and Appendix I sweeps the probe layer ℓ^*. Two further choices enter through the offline calibration rather than the method itself: the percentile rule that places the ramp endpoints $(\lambda_{\text{lo}}, \lambda_{\text{hi}})$, and the size of the calibration set. Table 7 varies one at a time on OLMoE-1B-7B with K and β fixed at their main-table values. When the calibration set changes, the complete procedure is rerun on it, including probe-layer selection by λ-AUROC and endpoint calibration; when only the percentile rule changes, the layer and calibration set stay as in the main table. Average safety is the mean over the six attacks; calibration size is the total number of held-out prompts, balanced between risky and benign.

Table 7: Sensitivity to the two offline calibration choices, on OLMoE-1B-7B. Default settings are marked; each block varies one factor with everything else held fixed.

Calibration choice	Setting	Avg. safety (↑)	XSTest (↑)
Ramp percentiles (benign / harmful)	90 / 10	46.5	95.0
	95 / 5 (default)	46.5	97.0
	97.5 / 2.5	46.0	98.0
Calibration-set size	50	45.8	96.0
	100	46.7	96.0
	200 (default)	46.5	97.0

Average safety moves by at most 0.9 points across all six settings. The percentile rule behaves as the design predicts: a more conservative endpoint (97.5 / 2.5) gates more queries out of the intervention and trades 0.5 safety points for 1 point of XSTest, while a more aggressive one (90 / 10) lowers XSTest by 2 points with no safety gain. Shrinking the calibration set fourfold, from 200 to 50 prompts, changes safety by 0.7 points, so the endpoints do not need a large held-out split to be placed usefully. Together with the K, β, and ℓ^* sweeps, this indicates that adapting PIRA to a new backbone is a lightweight, reproducible calibration rather than an unconstrained hyperparameter search.

E Judge Configuration

We follow PandaGuard's (Shen et al., 2025) LLM-judge methodology rather than lexical refusal-string matching, since string matching misses partial compliance and surface-paraphrased refusals. We adopt DeepSeek-V4-pro as the single judge for all methods on all backbones in this paper, and keep PandaGuard's rubric prompts unchanged so that the scores remain comparable to that protocol. Appendix F reports the same responses re-scored by a judge from a different family.

Safety prompt. For the six jailbreak attacks in Table 1, the judge sees the harmful goal and the model's response and emits an integer rubric score $r \in \{0, 1, \ldots, 100\}$, anchored at $r{=}0$ = "promotes harm, violence, exploitation," $r{=}50$ = "neutral," and $r{=}100$ = "helpful, ethical, promotes wellbeing," with the prompt explicitly telling the judge that refusals and helpful ethical advice both score 90–100. We then rescale to an alignment score $s = r/100 \in [0, 1]$ and label a response SAFE when $s \geq 0.7$. The threshold 0.7 (clearly leaning toward refusal rather than partial compliance) is stricter than thresholds commonly used in attack-side leaderboards, reflecting that we are scoring a refusal-side intervention, where partial compliance should not count as safe.

Over-refusal prompt. For XSTest, the same judge sees the benign-but-superficially-risky prompt and the response, and emits a binary label $\ell \in \{\text{ANSWER}, \text{REFUSE}\}$, anchored at ANSWER = "the assistant attempted to address the question (even partially)" and REFUSE = "the assistant declined, apologized, warned, or redirected away." The prompt explicitly informs the judge that the question is benign so that the judge does not penalise the assistant for engaging with surface-risky language. The label is parsed from the judge's first matched token via a case-insensitive regex matching either `ANSWER` or `REFUSE`; the XSTest answer rate is the fraction of responses with $\ell = \text{ANSWER}$.

Relationship to direction construction. The same judge produces the safe-versus-harmful partition that defines $\mathbf{d}_{\text{safe}}$ in Equation 3. Direction construction and evaluation therefore share an evaluator, which raises the question of how much of the measured gain reflects agreement with this judge's preferences rather than safer behavior. Two things bound the concern. The judge touches only the offline partition: no test prompt, no test response, and no test score enters the extraction of $\mathbf{d}_{\text{safe}}$ or the calibration of the ramp. And re-scoring every generated response with GLM-5.2 (Appendix F), which took no part in constructing the direction, preserves both the ranking and most of the margin. What this does not establish is that a direction extracted under a different labeling judge would select the same experts, which would require repeating the offline stage per judge; we leave that open.

Cross-method fairness. The same judge model, prompt templates, and threshold are applied identically to *Original*, *SteerMoE*, and PIRA on every backbone. Cross-method differences therefore reflect the methods themselves rather than any judge-side asymmetry.

F JUDGE ROBUSTNESS

A single judge family could in principle favor the method whose responses most resemble its own alignment style. We therefore re-score the *already generated* responses of all three methods on OLMoE-1B-7B with GLM-5.2, a judge from a different family, using the identical prompt templates and threshold. No model responses are regenerated, so any difference is attributable to the judge alone; the DeepSeek-V4-pro row reproduces the main-table values by construction.

Table 8: Average safe rate (%, ↑) over the six attacks on OLMoE-1B-7B under two judge families. Responses are identical across rows; only the judge changes.

Evaluation judge	Original	SteerMoE	PIRA
DeepSeek-V4-pro (used throughout the paper)	25.3	31.7	46.5
GLM-5.2	28.2	35.6	49.1

GLM-5.2 is the more lenient of the two, reading 2.6 to 3.9 more points of safety into the same responses for every method. The ordering PIRA $>$ SteerMoE $>$ *Original* and the size of the margin survive the change: PIRA leads SteerMoE by 14.8 points under DeepSeek-V4-pro and by 13.5 under GLM-5.2. A judge that played no part in constructing $\mathbf{d}_{\text{safe}}$ therefore sees the same effect.

G IMPLEMENTATION DETAILS

Router hook. OLMoE's gate module returns either a raw logit tensor (prefill) or a $(p, \text{top_val}, \text{top_idx})$ tuple (decode with KV cache). Our hook handles both by recomputing the bias addition before softmax and re-extracting top-k.

Decoding config. Greedy decoding throughout (`do_sample=False`, `max_new_tokens=256`, bf16); `pad_token_id` is set to `eos_token_id`. The same decoding config is applied to *Original*, *SteerMoE*, and PIRA to ensure cross-method comparability.

Direction normalization. Both offline directions $\mathbf{d}_{\text{risk}}$ and $\mathbf{d}_{\text{safe}}$ are unit-normalized per layer (each $\mathbf{d}_{\text{risk}}[\ell]$ and $\mathbf{d}_{\text{safe}}[\ell]$ is divided by its ℓ_2 norm) and stored once per backbone. Normalization removes the magnitude of the mean-difference vector; the projections $\lambda(q)$ and $s(q)$ still depend on the scale of $h(q)$.

Calibration of the modulation ramp. The endpoints $(\lambda_{\text{lo}}, \lambda_{\text{hi}})$ are set per backbone on a held-out split disjoint from the test prompts. We define λ_{lo} as the 95th percentile of $\lambda(q)$ on the benign half and $\lambda_{\text{hi}}^{(\text{harm})}$ as the 5th percentile on the risky half, then take $\lambda_{\text{hi}} = \max\big(\lambda_{\text{hi}}^{(\text{harm})}, \lambda_{\text{lo}} + \delta\big)$ with δ equal to one-tenth of the benign IQR. The max guarantees $\lambda_{\text{lo}} < \lambda_{\text{hi}}$ even when the benign and risky λ-distributions overlap heavily (i.e., when $\lambda_{\text{hi}}^{(\text{harm})}$ would invert below λ_{lo}): in that regime the ramp degenerates to a near-step at λ_{lo}, and the 95%-of-benign-below-λ_{lo} guarantee is preserved by construction. Calibration is one-shot at offline time and does not depend on the test query.

Compute. All experiments run on a single NVIDIA B200 (180 GB). Offline extraction of $\mathbf{d}_{\text{risk}}$ and $\mathbf{d}_{\text{safe}}$ takes $\sim$10 minutes per backbone at N=200 contrast pairs (collected once per backbone, distinct from the n=100 main-table evaluation cells). The online phase adds one forward-backward pass per query before generation, accounting for $\sim$5–10% of the cost of 256-token greedy generation (Section 3.4.3). End-to-end main-table generation takes $\sim$35 minutes per 100-query evaluation cell, and judging takes $\sim$5 minutes per 100 responses. Total compute for the full experimental sweep across three backbones, six attacks, three baselines, and the K / β / ℓ^* sweeps fits in under 200 B200-hours, putting the entire pipeline within reach of a single-GPU node.

H Per-Model Architecture and Hyperparameters

Table 9 summarizes the three MoE backbones used in the main results, together with Qwen3-30B-A3B from Appendix N. The probe layer ℓ^* is selected per backbone by the criterion of Section 3.4, the held-out λ-AUROC over a sparse grid of layers (Appendix I). Across all three backbones the selected layer falls in a comparable depth fraction ($\sim$0.55–0.65 of L), consistent with the "mid-layer plateau" picture and supporting the architecture-agnostic reading of the criterion. Qwen3-30B-A3B uses the default K=25 and a probe at layer 24 of 48. The bias strength β is shared across backbones, K=25 except on Mixtral (below), and the ramp endpoints are calibrated per backbone with the procedure of Appendix G.

Table 9: Architecture and selected hyperparameters per backbone. Depth fraction $= \ell^*/L$.

Model	L	E	top-k	hidden d	ℓ^*	depth frac.	K
OLMoE-1B-7B-0924-Instruct	16	64	8	2048	10	0.625	25
Qwen1.5-MoE-A2.7B-Chat	24	60	4	2048	14	0.583	25
Mixtral-8x7B-Instruct-v0.1	32	8	2	4096	20	0.625	4
Qwen3-30B-A3B (Appendix N)	48	128	8	2048	24	0.500	25

For Mixtral, K=4 is reduced from 25 to preserve a comparable density of intervened pairs across backbones. The eligible pool is the set of (layer, expert) pairs upstream of the probe, $\ell^* \cdot E$, so $K/(\ell^* \cdot E)$ is $25/640 \approx 3.9\%$ on OLMoE, $25/840 \approx 3.0\%$ on Qwen, and $4/160 \approx 2.5\%$ on Mixtral, all within the same order of magnitude. This is the natural cross-architecture calibration when K is interpreted as a global budget (Equation 7). The bias strength β and the calibration procedure for $(\lambda_{\text{lo}}, \lambda_{\text{hi}})$ are unchanged.

I Probe-Layer Sweep

We report the full probe-layer sweep referenced in Section 5.2 (probe-layer selection paragraph). The calibration set has 200 prompts (100 risky drawn proportionally across the six engaged at-

tacks, 100 benign drawn from XSTest), all disjoint from the test prompts in Table 1. We sweep $\ell^* \in \{4, 7, 10, 13, 15\}$ on OLMoE-1B-7B (a sparse five-point grid placed at depth fractions $\{0.25, 0.44, 0.625, 0.81, 0.94\}$, chosen to cover early, mid, and late blocks while keeping the calibration cost bounded). For each ℓ^* we report (i) the AUROC of $\lambda(q) = h(q) \cdot \mathbf{d}_{\text{risk}}[\ell^*]$ in classifying risky vs benign on the calibration set, (ii) the downstream safe rate (averaged over the six engaged attacks) when PIRA uses identification gradients taken at ℓ^*, and (iii) the XSTest answer rate when modulation uses $\lambda(q)$ at ℓ^*. All other hyperparameters are held fixed.

Table 10: Probe-layer sweep on OLMoE-1B-7B (calibration set, n=200). Identification (safe rate) saturates across mid-layers; modulation (XSTest, λ-AUROC) is peaked at $\ell^* = 10$. The layer is selected by λ-AUROC.

ℓ^*	λ-AUROC	Safe rate (↑)	XSTest (↑)
4	0.42	38	85
7	0.55	44	93
10	0.61	48	98
13	0.49	47	94
15	0.38	42	88

The two columns "safe rate" and "XSTest" track the two channels through which ℓ^* enters the method. Safe rate mainly reflects the quality of the identification gradient. XSTest captures the quality of $\lambda(q)$ as a benign-vs-risky gauge, since on benign prompts the identified set is irrelevant once $\lambda(q)$ falls below λ_{lo}. Identification benefits from a deeper probe: safe rate rises from 38% at $\ell^* = 4$ to 48% at $\ell^* = 10$ and saturates across $\ell^* \in \{10, 13\}$, consistent with a deeper probe attributing over more layers until the top-K=25 budget saturates. Modulation is sharply peaked in the middle: XSTest reaches 98 at $\ell^* = 10$ and drops at both ends (85 at $\ell^* = 4$, 88 at $\ell^* = 15$). Early layers have not yet encoded an abstract risk distinction, while the final layer specializes to generation rather than carrying it. The layer is chosen by the λ-AUROC column, which also peaks at $\ell^* = 10$.

J Statistical Reliability across Seeds

The main table reports single greedy runs, which removes sampling noise but leaves open whether the ordering survives stochastic decoding. We therefore re-run every method under stochastic decoding with temperature 0.7 and three independent generation seeds, holding all other settings fixed; the seeds are used solely to report the mean and standard deviation of each safe rate. For significance we use a goal-level paired sign-flip permutation test per backbone: for each of the 100 JBB goals we compute the PIRA minus SteerMoE difference in per-goal safe rate, averaged over the six attacks and the three seeds, and randomly flip the signs of the 100 paired differences (10,000 permutations, two-sided). Pairing at the goal level accounts for the correlation among attack variants of the same goal.

Table 11: Average safe rate (%, ↑) over the six attacks under stochastic decoding, mean ± standard deviation over three generation seeds. Absolute values differ slightly from Table 1, which uses greedy decoding.

Backbone	Original	SteerMoE	PIRA	PIRA − SteerMoE
OLMoE-1B-7B	24.9 ± 0.8	32.1 ± 1.2	45.8 ± 0.9	+13.7
Qwen1.5-MoE-A2.7B	25.1 ± 1.1	31.4 ± 0.7	45.6 ± 1.3	+14.2
Mixtral-8x7B	28.8 ± 0.6	37.2 ± 1.0	51.6 ± 0.8	+14.4

Run-to-run variation is small relative to the effect: standard deviations are 0.6–1.3 points while the gap over SteerMoE is 13.7–14.4 points, and the permutation test rejects the null on every backbone (largest $p = 0.0027$ across the three tests).

K Local Router Attribution: Derivation and Scope

Differentiation within a routing region. Fix a layer and token, and write A for its active top-k set, z_i for a router logit, and f_i for the expert output. For the top-k renormalization assumed in Equation 1, the mixture is

$$\tilde{p}_i = \frac{\exp z_i}{\sum_{j \in A} \exp z_j}, \qquad m = \sum_{i \in A} \tilde{p}_i f_i. \tag{10}$$

Away from routing ties, A is fixed in a neighborhood. Holding the block input fixed, its local derivative is

$$\frac{\partial m}{\partial z_e} = \tilde{p}_e (f_e - m), \qquad \frac{\partial s}{\partial z_e} = \tilde{p}_e g^\top (f_e - m), \quad e \in A, \tag{11}$$

where $g = \partial s / \partial m$ includes the downstream computation. For $e \notin A$ the local derivative is zero. Thus the derivative measures an expert's contribution relative to the current mixture. The calculation never differentiates through the discrete choice of experts, so it says nothing about what happens when a bias moves an expert across a routing boundary. The zero-gradient statement depends on top-k renormalization: if selected weights instead retain the full softmax denominator, inactive logits can affect that denominator. The general bias-gradient definition in Equation 6 must follow the backbone's actual forward computation.

Position and layer scope. Because a bias is shared across the prompt's token positions, the chain rule gives

$$\Delta_{\ell,e}(q) = \sum_{t=1}^{T(q)} \frac{\partial s(q)}{\partial \Psi'_{\ell,t,e}} \bigg|_{\Delta b = 0}. \tag{12}$$

Only positions with a computational path to the probe contribute. Under the input-state indexing of Section 3.1, layers $\ell \geq \ell^*$ are downstream of $h_{\ell^*,T(q)}$ and have zero attribution. Activations at earlier prompt positions can affect the last-token state through subsequent attention. This is prompt-side attribution; it does not measure the expert's contribution on future generated tokens.

First-order ascent of the safety score. For a fixed query, let b denote the complete deployed bias vector and $s_q(b)$ the prompt-side probe score under that bias. Freeze the set, gate, and weights computed at $b = 0$. If $\mathcal{A}^-$ is nonempty, define $M = \max_{i \in \mathcal{A}^-} |\Delta_i| > 0$, with i indexing (layer, expert) pairs. Equation 8 yields $b_i = \beta \rho \Delta_i / M$ for $i \in \mathcal{A}^-$, and zero otherwise. Consequently,

$$\frac{d}{d\alpha} s_q(\alpha b) \bigg|_{\alpha = 0} = \nabla s_q(0)^\top b = \frac{\beta \rho}{M} \sum_{i \in \mathcal{A}^-} \Delta_i^2 \geq 0. \tag{13}$$

For $\beta > 0$, $\rho > 0$, and a nonempty set, this derivative is positive. Where the original computation is differentiable, sufficiently small positive α therefore increases the safety score. If the gate is closed or the set is empty, $b = 0$. This result concerns a local direction, not the finite step $\alpha = 1$ or the probability of generating a safe response.

Single-expert ablations. Figure 3(c) tests the ranking in the regime the derivation does not cover. For each of 50 risky queries and each of the top-25 pairs in $\mathcal{A}^-(q)$, we apply to that pair alone the bias it receives in the deployed intervention (Equation 8), leave every other router logit unbiased, and regenerate the full response with the main-table decoding configuration. The judge scores the regenerated response on the same 0–100 alignment rubric used throughout (Appendix E), and the plotted quantity is the increase over the score of the unablated response to the same query. Since the bias has the deployed magnitude, it can move the expert out of the top-k set and admit a replacement, so each point measures a finite, routing-changing intervention rather than a derivative. Across the 1,250 pairs, the Spearman correlation between gradient rank and score increase is -0.81.

Finite intervention and generation. A finite bias can change top-k membership, admitting replacement experts whose outputs were not evaluated by the original local derivative. Even without a routing change, curvature can invalidate the first-order prediction at a large step. The bias magnitude for an individual expert is $\beta \rho w_i$, and displacement depends on competing biased logits at each token. Finally, s_q is a linear score fitted from observational continuation labels, and the bias derived from the prompt is reused throughout decoding; Section 5 tests both choices. One consequence of the derivation is worth stating: a small positive bias on a positive-gradient expert also raises the safety score, so the sign of the gradient alone does not account for the failure of the amplification variant in Section 5.2.

L Comparison with Weight-Tuning, Activation-Level, and Decoding-Level Alignment

SteerMoE is the only published router-level comparison and is used in Table 1 for that reason. The broader question is how a router-level intervention relates to alignment applied at other stages of the model. We evaluate two representative methods on Mixtral-8x7B with all PIRA settings unchanged: CAA (Rimsky et al., 2024) at the activation level, for which we independently select its intervention layer and multiplier on held-out calibration data, and RAIN (Li et al., 2024) at the decoding level. For PIRA + CAA, both interventions are applied during generation with the settings above. For PIRA + RAIN, PIRA computes its query bias from the original prompt and applies it only during RAIN's candidate-generation calls; RAIN's self-evaluation and search procedure are left unchanged. Because rewinding changes the generated tokens rather than the input query, the PIRA bias does not need to be recomputed.

Table 12: Alignment at different stages of the model, on Mixtral-8x7B. Average safe rate over the six attacks, XSTest answer rate, and MMLU 5-shot accuracy.

Method	Intervention level	Avg. safe rate ↑	XSTest ↑	MMLU ↑
Original	None	29.5	97.0	69.5
SteerMoE	Router	36.8	85.0	67.0
CAA	Activation	41.8	90.0	68.9
RAIN	Decoding	44.7	93.0	69.2
PIRA	Router	52.5	96.0	69.2
PIRA + CAA	Router + activation	56.2	90.0	68.7
PIRA + RAIN	Router + decoding	**59.0**	94.0	69.0

In Table 12, PIRA outperforms both on its own, and the two add a further 3.7 and 6.5 safety points on top of it while carrying over their characteristic helpfulness costs. Appendix N adds the remaining comparison, against a method that tunes expert weights offline.

M Generalization to Unseen Risk Categories

Table 3 tests attack-level generalization: the risk taxonomy is JBB's throughout, and only the attack template changes. A stronger shift is to change the taxonomy itself. We use SORRY-Bench (Xie et al., 2025), whose categories are balanced by construction, and select seven categories that are not represented as top-level categories in JBB: Self-Harm, Child-Related Crimes, Animal-Related Crimes, Environmental Crimes, Intellectual Property Infringement, Evasion of Law Enforcement, and Public Order Crimes. For each category we evaluate its ten base instructions under all 20 official linguistic mutations, giving 200 prompts per category. No SORRY-Bench data is used at any offline step: the directions, the ramp endpoints, and the SteerMoE expert pool all remain the ones extracted from the JBB suite on OLMoE-1B-7B. We score with SORRY-Bench's official Mistral-7B evaluator and report one minus its compliance rate as the safe-refusal rate.

Table 13: Safe-refusal rate (%, ↑) on seven SORRY-Bench categories held out from every offline step, on OLMoE-1B-7B. IP = Intellectual Property, LE = Law Enforcement.

Method	Self-Harm	Child	Animal	Environ.	IP Infring.	LE Evasion	Public Order	Avg.
Original	46	58	62	54	48	44	50	51.7
SteerMoE	56	64	70	60	**58**	56	62	60.9
PIRA	**65**	**71**	**75**	**68**	57	**64**	**70**	**67.1**

PIRA improves over SteerMoE in six of the seven held-out categories and raises the average by 6.2 points; on IP Infringement it stays within one point of SteerMoE. The margin is smaller than on the JBB evaluation, as expected under this stronger distribution shift.

N A Newer-Generation MoE Backbone

The three main backbones are drawn from an earlier generation of open MoE models with relatively low base safety, which leaves room for any intervention to improve. We therefore repeat the evaluation on Qwen3-30B-A3B (Yang et al., 2025), whose base safe rate on the same six attacks is already 76.2, using the identical protocol and, for SteerMoE, its authors' released Qwen3 configuration (Table A.2 of Fayyaz et al. (2026)). Table 14 reports the result and also places PIRA against RASA (Liang et al., 2026), which fine-tunes safety-critical experts offline; we use its released implementation and recommended training recipe. For PIRA + RASA we first complete RASA training, then re-extract PIRA's directions and recalibrate its thresholds on the resulting checkpoint, with all RASA-aligned weights frozen thereafter.

Table 14: Qwen3-30B-A3B, a newer-generation MoE backbone with high base safety. Average safe rate over the six attacks, XSTest answer rate, and MMLU 5-shot accuracy.

Method	Offline weight tuning	Avg. safe rate ↑	XSTest ↑	MMLU ↑
Original	No	76.2	**98.0**	**79.5**
SteerMoE	No	84.0	94.0	78.9
RASA	Yes	93.5	96.0	79.1
PIRA	No	89.2	97.0	79.3
RASA + PIRA	RASA only	**95.8**	95.0	79.0

The per-input routing principle transfers to the newer backbone: PIRA raises the safe rate by 13.0 points over *Original* and by 5.2 over SteerMoE while retaining a 97.0 XSTest answer rate and moving MMLU by 0.2 points. The absolute margin over SteerMoE is smaller than in Table 1 because this backbone starts from substantially higher base safety. The comparison with RASA, by contrast, is one between different budgets rather than a like-for-like one. RASA attains 4.3 points more raw safety than PIRA by training the router and the safety-critical experts, whereas PIRA reaches 13.0 points over *Original* without touching any weight and at a 1-point XSTest cost. The two are complementary: applying PIRA on top of a RASA-aligned checkpoint adds a further 2.3 safety points for a 1-point XSTest change and a 0.1-point MMLU change, because RASA aligns the weights offline while PIRA adds the query-specific routing decision at inference.

O Inference Cost, Batching, and Long Contexts

Section 3.4.3 summarizes PIRA's cost; this appendix reports the measurements behind it. All numbers are on a single NVIDIA B200 with Qwen3-30B-A3B in BF16, non-thinking mode, greedy decoding, and `max_new_tokens=256`. We use the larger backbone here because it is the least favorable of the four for a per-query probe.

Where the cost lands. PIRA performs one probe forward-backward pass to obtain the per-query bias, then a fresh biased prefill followed by ordinary autoregressive decoding. No backward pass is performed during decoding, so the cost is a fixed prompt-side addition rather than a per-token tax.

Table 15: Latency breakdown for a single query on Qwen3-30B-A3B, 256-token completion. TTFT is time to first token; *Decoding* is the time spent after the first token.

Method	Probe F+B (s)	Prefill (s)	Decoding (s)	TTFT (s)	Total (s)	Overhead
Original	–	0.26	12.43	0.29	12.72	–
PIRA	1.13	0.27	12.44	1.43	13.87	9.0%

The biased prefill costs essentially the same as the original prefill (0.27 versus 0.26 s) and decoding is unchanged (12.44 versus 12.43 s), so the entire difference is the one-time probe. It raises user-visible TTFT by 1.14 s and amortizes to 9.0% over a 256-token completion. Interactive deployments that stream short answers therefore see the probe as added TTFT rather than as a throughput cost.

Batched serving. PIRA does not need a separate backward pass per query in a batch. We assign each example a batch-indexed router-bias tensor of shape $[B, L, E]$, perform one truncated batched

forward pass, and backpropagate the sum of the B safety scores. Because the per-example scores do not interact, each row of the resulting gradient is the attribution for the corresponding request, so all per-query attributions are obtained in a single batched forward-backward pass. The bias itself is cheap to represent, only $B \times L \times E$ scalars; the main cost of batching is the activation memory retained for the probe backward.

Table 16: Batched execution on Qwen3-30B-A3B with 256-token prompts and 256-token completions. *Seq.* probes each query separately before batched generation; *batched* computes all probes jointly. Peak memory is reported as *Original* / PIRA.

Batch	Original (q/s)	Seq. PIRA (q/s)	Batched PIRA (q/s)	Batched / Orig.	Peak mem. (GiB)
1	0.0786	0.0721	0.0721	91.7%	65.4 / 73.8
2	0.146	0.126	0.135	92.5%	70.8 / 84.9
4	0.259	0.205	0.237	91.5%	82.7 / 111.6
8	0.411	0.282	0.369	89.8%	107.3 / 158.2

Batched and sequential probing select the same top-K expert pairs on 98.8% of pairs, with the remaining disagreements at near-tied router scores and no measurable change in aggregate safety. Batching improves probe utilization, so PIRA retains 89.8–92.5% of the original throughput at batch sizes 1–8; the modest decline at batch 8 comes from the token-proportional backward workload and the increased activation memory, and at that batch size batched probing is still 30.9% faster than probing each query in turn. Measured instead as the turnaround for a fixed set of $N{=}16$ requests (128-token prompts, 256-token completions) on the same backend, *Original* and PIRA take 146.63 and 151.21 s at batch size 1, 36.93 and 38.15 s at batch size 4, and 9.21 and 9.48 s at batch size 16, an overhead of 2.9–3.3%; the probe's latency stays amortized over the completion while its activation memory grows with the probe batch.

The probe's activation memory grows with batch size; its time does not. The memory can be bounded independently of the generation batch by splitting the probe into microbatches. Under a stress test at batch size 512 (128-token prompts, 128-token completions), reducing the probe microbatch from 256 to 64 lowers PIRA's peak memory from 152.3 to 80.8 GiB, against 71.9 GiB for *Original*, while increasing turnaround time by only 2.4 percentage points (31.2% to 33.6% overhead).

Long contexts. The probe's cost grows with prompt length, but its forward-backward pass is structurally lighter than a full training step and does not retain the complete activation graph. It stops at layer 24 of Qwen3's 48 and follows only the eight experts activated per token at each probed MoE layer; it backpropagates into the router-bias variables alone with all model weights frozen, so it propagates activation gradients through the active path but computes no parameter gradients; and it uses checkpointing at layer boundaries, recomputing intermediate states during the backward. The truncated sparse probe runs on SonicMoE (Guo et al., 2026), an IO- and tile-aware implementation of memory-efficient sparse-MoE forward and backward passes, whereas Tables 15 and 16 use the Hugging Face backend for both probe and generation; the probe timings in the two settings are therefore not directly comparable. These choices do not remove the dependence on context length, but they reduce its magnitude. To measure the resulting profile under a realistic serving path, we run the two stages sequentially, $T_{PIRA} = T_{\text{vLLM}} + T_{\text{probe}}$, generating 256-token completions with vLLM (Kwon et al., 2023) compilation and CUDA graphs enabled.

Table 17: Long-context cost on Qwen3-30B-A3B: vLLM generation of a 256-token completion plus the standalone PIRA probe.

Input length	Original vLLM (s)	PIRA probe (s)	Total (s)	Overhead	Probe activations (GiB)
2,048	0.783	0.154	0.937	20%	1.20
4,096	0.787	0.199	0.986	25%	1.55

Doubling the input from 2K to 4K increases probe time by 29% rather than twofold, and total overhead by 5 percentage points. At 4K the one-time 0.199 s probe is set against 0.787 s of generation, which covers the full 48-layer prefill and 256 sequential KV-cached decoding steps. Checkpointing keeps the memory flat: without it, probe activations at 4K reach 8.63 GiB, against 1.55 GiB with it, a 5.6×

reduction bought with 21% additional probe time ($0.164\,\mathrm{s}$). Context length does increase PIRA's cost, and the probe is least amortized when a long prompt is answered in few tokens.

On the choice of backend. The probe needs autograd over per-request router state, which current high-throughput engines do not expose: vLLM's main GPU path runs under `torch.inference_mode()` and its Transformers fallback still uses a non-differentiable cache, and SGLang (Zheng et al., 2024) (v0.5.9) runs the original model with continuous batching but executes its Transformers worker without gradients and does not expose differentiable per-request router state either. Tables 15 and 16 therefore run *Original* and PIRA on the same autograd-capable backend, which isolates PIRA's own time and memory cost; comparing a vLLM or SGLang *Original* against a Hugging Face PIRA would instead conflate the intervention with fused kernels, paged attention, CUDA graphs, and scheduler differences. A native integration requires a separate autograd-enabled probe worker and per-request router-bias metadata for packed tokens, which we leave as future work.