Steering Out Position Bias: A Probe-Localized, Single-Layer Intervention for Pairwise LLM-as-a-Judge

Anonymous ACL submission

Abstract

LLM-as-a-Judge has emerged as a scalable alternative to costly human evaluation. However, LLM judges suffer from position bias in pairwise evaluation: the judge systematically favors the first or second candidate regardless of content, forcing practitioners to perform bidirectional evaluation at twice the inference cost. We propose **Probe-and-Steer (PaS)**, a probe-localized, single-layer steering method for mitigating position bias. PaS localizes the bias-critical layer via a layer-wise logit-difference probe and trains a single residual-stream steering vector at that layer under a multi-objective loss, requiring minimal labeled data while keeping the LLM fully frozen. Through layer-wise probing across multiple model families, we uncover three qualitatively distinct bias-distribution patterns and delineate a clear applicability boundary for residual-stream steering. Experiments on an XSum-based dataset with `Llama-3.1-8B-Instruct` show that PaS delivers the largest gains among all compared methods, improving consistency by 22.9 pp and accuracy by 16.4 pp while shrinking positional asymmetry by 33× (from 0.471 to 0.014), adding the lowest inference overhead at 4%.

1 Introduction

Large language models (LLMs) are increasingly deployed as automatic evaluators of text quality, a paradigm known as *LLM-as-a-Judge* (Zheng et al., 2023; Liu et al., 2023; Chiang and Lee, 2023). Dubois et al. (2023) report that pairwise LLM evaluation is approximately 45× cheaper than crowdsourcing ($70 vs. $3,150 per experiment) and shortens iteration from days to hours. This efficiency has made LLM-as-a-Judge a preferred choice for benchmark construction (Zheng et al., 2023; Dubois et al., 2023; Li et al., 2025) and reinforcement learning from AI feedback (Bai et al., 2022; Lee et al., 2024; Cui et al., 2024). The paradigm, however, hinges on a fragile assumption: that judgments are driven by content quality rather than surface cues such as candidate position.

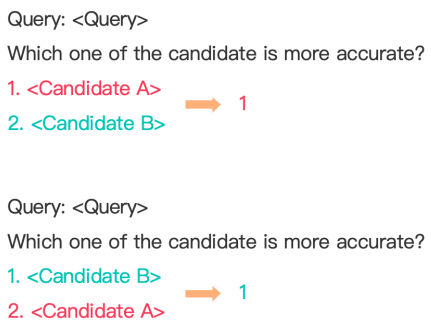

Figure 1: An example of position bias in pairwise LLM-as-a-Judge.

Recent work exposes *position bias*: the judge systematically favors the first or second candidate regardless of content (Figure 1) (Wang et al., 2024; Zheng et al., 2024; Pezeshkpour and Hruschka, 2024). The most systematic account comes from Shi et al. (2025), who probe 15 judges across 22 tasks and roughly 150,000 comparisons on MT-Bench and DevBench, reporting a consistency of only 0.69 for `Llama-3.1-8B` in pairwise evaluation. These results, however, come from public benchmarks with documented test-set contamination (Sainz et al., 2023; Golchin and Surdeanu, 2024), raising concerns that position-bias measurements may be partially masked by memorized verdicts. For a contamination-controlled view, we construct a new pairwise evaluation set from XSum articles (Narayan et al., 2018). On this set, `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024) produces content-consistent verdicts for only 50% of pairs, 19 pp below the 0.69 on MT-Bench (§4).

Such brittleness directly affects evaluation pipelines. To prevent position bias from contaminating model evaluation, major LLM-as-a-Judge protocols, including MT-Bench, AlpacaEval, and

Arena-Hard, adopt *bidirectional evaluation*, running the judge twice with swapped candidate orders and retaining consistent pairs (Zheng et al., 2023; Wang et al., 2024; Pezeshkpour and Hruschka, 2024; Shi et al., 2025; Li et al., 2024; Dubois et al., 2023; Li et al., 2025). This doubles inference cost and shrinks the effective evaluation set by discarding inconsistent instances, risking a distorted view of judge performance. These drawbacks motivate mitigating position bias in the judge itself.

Existing inference-time calibrations, including Attention Sorting (Peysakhovich and Lerer, 2023), PCW (Ratner et al., 2023), PINE (Wang et al., 2025), SteerFair (Adila et al., 2024), and the channel-scaling method of Yu et al. (2025), were not designed for pairwise evaluation. They originate from settings such as multi-document ranking and presuppose candidates of different relevance to the query, an assumption violated in pairwise evaluation where both candidates are equally task-relevant. Beyond this mismatch, four of them (excluding Attention Sorting) intervene broadly on model internal computations: PCW and PINE globally rewrite attention patterns and positional encodings, while SteerFair and channel-scaling modify attention-head activations or hidden-state channels across many layers (e.g., layers 10–25). Such pervasive, multi-site interventions perturb representations the model never encountered during pretraining and may degrade its general capabilities (Stickland et al., 2024; Hoelscher-Obermaier et al., 2023). Our experiments (§4) confirm that when transferred to pairwise evaluation, these methods either exacerbate position bias or yield limited mitigation. These limitations motivate a minimally invasive method tailored to the pairwise scenario.

We start from a structural property of judges: mechanistic probing shows that decision-relevant features such as truthfulness and style concentrate in a narrow band of mid-to-late layers rather than uniformly across depth (Tenney et al., 2019; Geva et al., 2022; Meng et al., 2022; Zou et al., 2023; Burns et al., 2023; Li et al., 2023; Rimsky et al., 2024). Position anchoring is precisely this kind of late-emerging shortcut. Representation engineering further shows that such behavioral tendencies are linearly encoded as residual-stream directions and can be modulated by simple vector addition (Zou et al., 2023; Turner et al., 2023; Rimsky et al., 2024). We therefore hypothesize that position bias concentrates at specific residual-stream depths and can be mitigated by a *single-layer* intervention. Editing the residual stream at the peak depth targets the bias where its signal is strongest, whereas spreading perturbations across multiple layers risks over-correction through coupled residual streams, a failure mode we confirm empirically (§B). Concretely, we first *locate* the layer at which positional preference peaks via a logit-difference probe, then *steer* the residual stream at that layer with a supervised vector trained under three complementary objectives. This requires only 100 labeled pairs and keeps the backbone fully frozen. Experiments reveal three distinct bias-distribution patterns that delineate a clear applicability boundary for PaS and show that, on an XSum-based dataset with `Llama-3.1-8B-Instruct`, PaS achieves the largest improvements among all baselines, improving consistency by 22.9 pp and accuracy by 16.4 pp while reducing positional asymmetry by $33\times$ (from 0.471 to 0.014).

To summarize, we:

1. We propose Probe-and-Steer (PaS), a probe-localized, single-layer residual-stream steering method for mitigating position bias in pairwise evaluation. Among all baselines, PaS delivers the largest consistency and accuracy gains while incurring the lowest inference overhead.
2. We reveal three qualitatively distinct layer-wise distributions of position bias via logit-difference probing and delineate a clear applicability boundary for residual-stream steering (§4.2, §5.2).

2 Related Work

2.1 LLM-as-a-Judge

Using LLMs to score or rank candidate outputs has emerged as a scalable proxy for costly human evaluation. Chiang and Lee (2023) demonstrate that strong LLMs reproduce human ratings on open-ended generation, and Liu et al. (2023) formalize this with G-Eval, a chain-of-thought protocol eliciting fine-grained scores from GPT-4. The pairwise variant, which selects the better of two candidates in a single prompt, was popularized by Zheng et al. (2023) through MT-Bench and Chatbot Arena. Open-source judges such as Prometheus (Kim et al., 2024) show that mid-sized LLMs can approach proprietary judges given suitable supervision, and the paradigm has expanded to benchmark construction and reinforcement learning from AI feedback (Bai et al., 2022). These studies establish the *utility* of LLM judges but implicitly assume that the judg-

ments are content-grounded. Our work **departs from** this assumption: we find that nearly half of all judgments produced by `Llama-3.1-8B-Instruct` reflect a position-1 shortcut rather than a genuine quality preference.

2.2 Position Bias and Mitigation Strategies

Position bias has been documented across model families and evaluation formats. Wang et al. (2024) show that flipping candidate order often flips the verdict, Zheng et al. (2024) report analogous fragility in multiple-choice settings, and Pezeshkpour and Hruschka (2024) confirm the pattern across diverse prompt formats. Shi et al. (2025) further demonstrates the persistence of position bias across 15 judges, 22 tasks, and approximately 150,000 comparisons. Existing remedies fall into four families. *Bidirectional evaluation* (Zheng et al., 2023; Wang et al., 2024) runs the judge twice with swapped orders, doubling the inference cost and risking a distorted view of the true performance of the judge. *Prompt engineering* restructures the input; for instance, Li et al. (2024) segment and realign candidate answers, but such approaches yield only marginal gains (Pezeshkpour and Hruschka, 2024). *Training-based* approaches modify model weights with position-agnostic objectives (He et al., 2024; Zhu et al., 2025) at substantial data and compute cost. *Inference-time calibrations* intervene in inference procedure without weight updates, but they vary in how precisely they localize the source of the bias. As shown in §4, several of these methods fail catastrophically in the pairwise setting. Unlike these approaches, which either modify the model's computations across many layers or only reorder prompt content without touching the model, PaS is the first probe-localized, single-layer residual-stream intervention, and it eliminates the need for a second forward pass.

2.3 Activation Steering

Activation steering intervenes directly on internal representations without parameter updates. Subramani et al. (2022) show that latent steering vectors extracted from frozen LMs can recover targeted generations, and Li et al. (2023) introduce Inference-Time Intervention (ITI), shifting attention-head activations along truthfulness directions. Zou et al. (2023) generalize this perspective as *representation engineering* and demonstrate that alignment-relevant properties correspond to low-rank residual-stream directions. More recently, Activation Addition (Turner et al., 2023), its contrastive extension (Rimsky et al., 2024), and In-Context Vectors (Liu et al., 2024) extend steering to behavior control and in-context learning. Despite proven effectiveness for truthfulness, style, and safety, its application to pairwise position bias has been limited to SteerFair (Adila et al., 2024), which applies unsupervised PCA on per-head activations to remove bias directions. Our work differs in three ways: (i) we *localize* the intervention to the bias-critical layer via probing rather than applying it uniformly across heads; (ii) we train the vector with a *supervised multi-objective loss* rather than unsupervised PCA; and (iii) we intervene on the *residual stream* with a single vector rather than on per-head outputs.

3 Method

We present **Probe-and-Steer (PaS)**, a probe-localized, single-layer residual-stream intervention for mitigating position bias in pairwise LLM-as-a-Judge. We first formalize position bias in the pairwise setting, then probe its signal at every layer, train a steering vector at the bias-critical layer under a joint loss, and apply it in a single forward pass at inference.

3.1 Problem Setup

We illustrate the formalization with pairwise summarization evaluation on XSum (Narayan et al., 2018); the construction transfers to all tasks in §4. A frozen judge J receives an article a and two candidate summaries s_A and s_B, and selects the better summary by producing the next token over option tokens "1" and "2." We define the *forward permutation* π as the configuration that places s_A at position 1 and s_B at position 2; the *backward permutation* π' is the reverse. The content of the pair is fixed; only spatial arrangement varies.

We define **position bias** as any change in judge output arising solely from candidate positions rather than content. Because π and π' swap candidates, a judge selecting the same *content* under both orders must output *different* option tokens: if the judge prefers s_A and outputs "1" under π, it must output "2" under π'. A content-grounded judge therefore satisfies $J(\pi) \neq J(\pi')$ for all pairs. A position-biased judge, by contrast, outputs the *same* token under both orders (e.g., always "1"), exhibiting a **structural asymmetry** independent of which content occupies that slot.

3.2 Probing for the Bias-Critical Layer

Because modifying all layers risks degrading performance by deviating from the original training distribution, we propose a steering-vector-based method that mitigates position bias with far less modification to the model than existing approaches. This raises a key design question: *at which layer should the steering vector be trained and injected?* To answer it, we apply a layer-wise logit-difference probe to every layer of the judge.

Using `Llama-3.1-8B-Instruct` (32 layers) as an example, the procedure is as follows:

- For each layer $\ell \in \{0, \ldots, 31\}$, we extract the last-token hidden state $\mathbf{h}_\ell \in \mathbb{R}^{4096}$ from the output of the ℓ-th transformer block during a standard forward pass.
- We project $\mathbf{h}_\ell$ through the unembedding matrix W_U of the model, read off the logits $z_\ell^{(1)}$ and $z_\ell^{(2)}$ for the option tokens "1" and "2," and compute the **logit difference** $d_\ell = z_\ell^{(1)} - z_\ell^{(2)}$.
- For each sample, we compute d_ℓ under both the forward (π) and backward (π') permutations and take the average: $\Delta_\ell = (d_\ell^{\pi} + d_\ell^{\pi'})/2$.
- Averaging $|\Delta_\ell|$ across 20 validation samples yields the **bias curve** $\overline{|\Delta|}_\ell$, plotted in Figure 2.

The **bias-critical layer** ℓ^* is defined as the layer at which $\overline{|\Delta|}_\ell$ peaks. For `Llama-3.1-8B-Instruct` on XSum-based pairs, ℓ^* corresponds to **layer 24** out of 32 (Figure 2). We argue that ℓ^* provides the maximal leverage for the intervention where positional preference is strongest and most concentrated. Our layer ablation (§4.5) confirms that L24 outperforms alternatives by up to +6.5 pp in consistency.

3.3 Steering Vector Training

At the bias-critical layer $\ell^* = 24$, we inject a learnable vector $\mathbf{v} \in \mathbb{R}^{4096}$ into the residual stream. The injection is applied only at the last-token position, which directly drives the next-token logits:

$$\mathbf{h}_{\ell^*}^{\text{new}} = \mathbf{h}_{\ell^*}^{\text{old}} + \alpha \cdot \mathbf{v} \tag{1}$$

where α denotes the steering strength. We train $\mathbf{v}$ under three complementary objectives, each targeting one facet of position bias.

$\mathcal{L}_1$**: Invariance Loss (Uniform-KL).** On *same-content pairs*, which are constructed by placing an identical summary at both positions, any deviation from a uniform output distribution must arise from a positional preference. We therefore minimize

$$\mathcal{L}_1 = \mathrm{KL}\big([P_1,\, P_2] \,\big\|\, [0.5,\, 0.5] \big)\,. \tag{2}$$

Same-content pairs isolate the positional signal, so $\mathcal{L}_1$ directly penalizes the position anchoring observed in vanilla judges.

$\mathcal{L}_2$**: Consistency Loss (MSE).** On *quality-difference pairs* presented in both orders, a content-grounded judge should assign the same probability to the better summary regardless of position:

$$\mathcal{L}_2 = \big(P(s_A \mid \pi) - P(s_A \mid \pi') \big)^2. \tag{3}$$

A perfectly bidirectionally consistent judge attains $\mathcal{L}_2 = 0$, selecting the same content regardless of position and eliminating the need for bidirectional inference.

$\mathcal{L}_3$**: Calibration Loss (Cross-Entropy).** Pure de-biasing risks degrading judgment quality. To address this risk, $\mathcal{L}_3$ preserves accuracy by anchoring the judge to ground-truth preferences under *both* permutations:

$$\mathcal{L}_3 = \mathrm{CE}([P_1, P_2]_\pi,\ y) + \mathrm{CE}\big([P_1, P_2]_{\pi'},\ y'\big)\,, \tag{4}$$

where y is the gold winner label (majority vote of Gemini 2.5 Pro (Gemini Team, Google, 2025), DeepSeek-V3 (DeepSeek-AI, 2024), and Claude Sonnet 4.5 (Anthropic, 2025)) and y' is its position-flipped counterpart. The bidirectional formulation maintains quality regardless of presentation order. $\mathcal{L}_3$ supplies the only content-grounded supervision: $\mathcal{L}_1$ and $\mathcal{L}_2$ shape position-invariant behavior regardless of correctness, whereas $\mathcal{L}_3$ ties the steered judge to ground-truth preferences under both orders. Our loss ablation (Appendix D) confirms that the full three-objective combination yields the most balanced trade-off across consistency, accuracy, and asymmetry.

Joint Loss. Beyond the three objectives above, we add an L_2 regularizer on $\mathbf{v}$ to minimize deviation from the original training distribution, yielding the full objective:

$$\mathcal{L} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \lambda_3 \mathcal{L}_3 + \lambda_{\text{mag}} \|\mathbf{v}\|_2^2. \tag{5}$$

We set $\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_3 = 2$, and $\lambda_{\text{mag}} = 0.01$. A sensitivity analysis (Appendix E) confirms the robustness of these choices: varying λ_3 across an $8\times$ range (from 0.5 to 4.0) changes consistency by no more than 0.7 pp.

Training Procedure. We train $\mathbf{v}$ on 100 training triplets with majority-vote preference labels from

Gemini 2.5 Pro, DeepSeek-V3, and Claude Sonnet 4.5 (§4.1). Each triplet contributes five signals: same-content pairs feed $\mathcal{L}_1$, while forward and backward permutations of the quality-difference pair feed both $\mathcal{L}_2$ and $\mathcal{L}_3$. Only 4,096 parameters are learned, and the 8B-parameter backbone remains frozen. We use AdamW (lr 10^{-4}, batch size 1, gradient accumulation 16, 50 epochs), and training converges in approximately 70 minutes on a single A100 GPU.

3.4 Inference-Time Application

At deployment, we apply $\alpha\mathbf{v}$ to the residual stream of layer 24 during a **single forward pass** (Eq. 1, $\alpha = 1.2$). No backward pass or post-hoc averaging is required. The overhead is a single 4,096-dimensional vector addition at one layer, negligible relative to the 8B-parameter forward pass. In terms of wall-clock time, PaS adds approximately 4% latency over vanilla inference on 140 samples (versus +100% for bidirectional evaluation); a detailed cost analysis is provided in Appendix A.

Remarks. We have presented a *probe-and-steer* pipeline that (i) localizes the bias-critical layer through forward and backward logit-difference probing, (ii) trains a 4,096-dimensional residual-stream steering vector under three complementary objectives that target invariance, consistency, and accuracy, and (iii) deploys this vector through a single forward pass at negligible cost.

4 Experiments

4.1 Experimental Setup

Configuration. We evaluate PaS on three tasks: XSum summarization (Narayan et al., 2018), NQ open-domain QA (Kwiatkowski et al., 2019), and MBPP code generation (Austin et al., 2021), across three open-source judges: `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024), `Qwen-2.5-7B-Instruct` (Yang et al., 2024), and `Mistral-7B-Instruct-v0.3` (Jiang et al., 2023). Each judge receives a task input and two candidate responses (one from Llama-3.1-8B-Instruct, one from GPT-3.5-Turbo (Ouyang et al., 2022)) and selects the better one through next-token prediction over tokens "1" and "2."

Ground-truth annotation. Preference labels are obtained by independently annotating each pair with Gemini 2.5 Pro (Gemini Team, Google, 2025), DeepSeek-V3 (DeepSeek-AI, 2024), and Claude Sonnet 4.5 (Anthropic, 2025), and then aggregating the three annotations by majority vote.

Data splits. From the annotated triplets, we construct non-overlapping partitions for each task: 100 training samples, 20 validation samples, and 140 test samples for XSum, 120 test samples for NQ, and 100 test samples for MBPP. The probing step uses the validation samples. The steering strength α is selected on the validation set by sweeping over $\alpha \in \{0.5, 0.8, 1.0, 1.2, 1.5, 2.0\}$, and the test set is evaluated only at the selected value of α. All evaluations use greedy decoding (temperature = 0) on a single A100 80 GB GPU.

Baselines. We compare PaS against five baselines: (i) Vanilla inference, (ii) Attention Sorting (Peysakhovich and Lerer, 2023), (iii) PCW (Ratner et al., 2023), (iv) PINE (Wang et al., 2025), (v) Positional Hidden Channel Scaling (Yu et al., 2025), and (vi) SteerFair (Adila et al., 2024). All methods share the same task-specific training set.

Evaluation Metrics. We adopt three metrics. (i) CONSISTENCY-RATE $r_{\text{con}} = \Pr[J(\pi) \neq J(\pi')]$: fraction of pairs whose option tokens differ across orders, measuring the consistency of the judge. (ii) ACCURACY-RATE $r_{\text{acc}} = \Pr[J(\pi) = y_\pi \wedge J(\pi') = y_{\pi'}]$: fraction of pairs where both orders correctly identify the winner, capturing content-grounded judgment quality. (iii) ASYMMETRY-INDEX $\mathcal{A} = \Pr[J(\pi) = J(\pi') = 1] - \Pr[J(\pi) = J(\pi') = 2]$: gap between position-1 and position-2 anchoring; a content-grounded judge satisfies $\mathcal{A} \approx 0$.

4.2 Layer-Wise Probing across Models

We apply the logit-difference probe (§3.2) to all nine model–task combinations. Figure 2 reveals three qualitatively distinct patterns of bias distribution across the model families.

Llama-3.1-8B: gradual-accumulation type. Position bias accumulates through the mid-to-deep transformer layers. This gradual distribution provides a wide target band for steering intervention.

Qwen-2.5-7B: deep-concentrated type. The bias concentrates sharply at layers 25–26 before dropping abruptly at the final layer. The bias direction in Qwen is *negative* ($d_\ell < 0$, favoring position 2), which is opposite to that of Llama and Mistral.

Mistral-7B: last-layer spike type. Layers 0–30 exhibit near-zero bias, followed by a massive spike at layer 31. This pattern indicates that position bias in Mistral is introduced almost entirely at the final layer rather than being accumulated through the

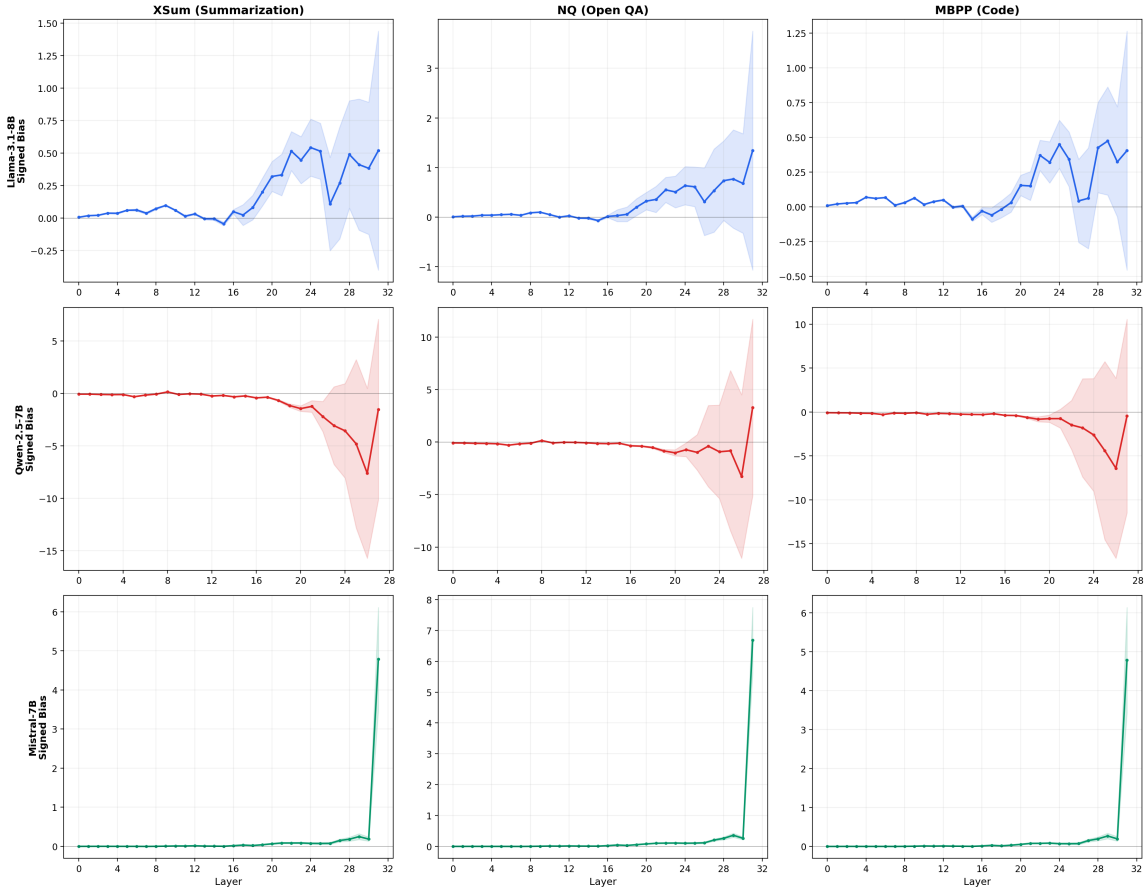

Figure 2: Signed per-layer logit difference for all nine model–task combinations (20 probing samples $\times$ 2 directions; mean $\pm$ std). Llama-3.1-8B (top row) exhibits a *gradual-accumulation* pattern that peaks at L24 on XSum; Qwen-2.5-7B (middle row) shows a *deep-concentrated* negative spike at L25–26; and Mistral-7B (bottom row) exhibits a *last-layer spike* at L31.

intermediate layers.

Cross-model implications. These three patterns have direct consequences for the feasibility of steering. The gradual-accumulation pattern offers a wide intervention window at mid-depth layers where bias is strong but not yet entangled with output logits. The last-layer spike pattern, in contrast, concentrates all bias immediately preceding the unembedding, where residual-stream perturbations cause unstable logit oscillations and gradient optimization converges to near-zero vectors. We analyze this failure mode in §5.

4.3 Main Results of Effectiveness

Baseline comparison on XSum. Figure 3 reports results on the XSum-based summarization test set with `Llama-3.1-8B-Instruct`. PaS achieves the best performance, improving consistency by 22.9 pp and accuracy by 16.4 pp while reducing asymmetry from 0.471 to 0.014 (33$\times$ reduction), indicating near-symmetric residual bias.

Multi-task and multi-model effectiveness. To assess the effectiveness of PaS across different models and tasks, we replicate the full probe-and-steer pipeline on all nine model–task combinations (Figure 4).

PaS delivers double-digit consistency gains on Llama-3.1-8B across all tasks (+15.8 pp to +22.9 pp), confirming task-agnostic effectiveness. Qwen-2.5-7B exhibits more modest gains (+5.7 pp to +11.3 pp), while Mistral-7B shows no response on any task, directly predicted by its last-layer spike pattern §4.2.

4.4 Cross-Task and Cross-Model Transfer

To assess generalization, we transfer steering vectors across tasks and models *without retraining*.

Cross-task transfer. On Llama-3.1-8B, all six cross-task transfer directions yield positive consistency improvements (from +4 pp to +19.3 pp; Table 1), which indicates that the position-bias direction in the residual stream shares common sub-

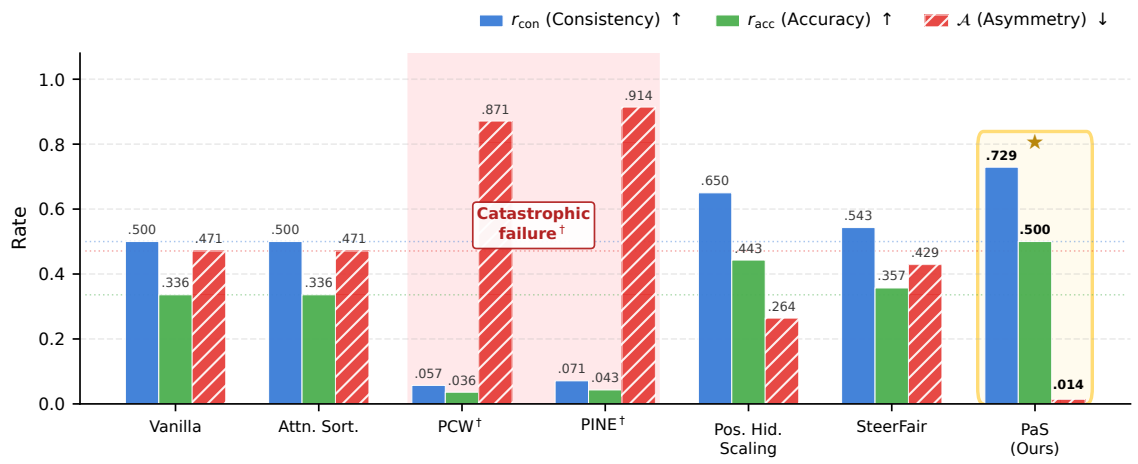

Figure 3: Main results on XSum with `Llama-3.1-8B-Instruct` ($N = 140$ test, greedy decoding). The dotted lines indicate the Vanilla references. PaS (gold frame, ★) achieves the best performance on every metric and reduces the asymmetry by a factor of 33 relative to Vanilla.

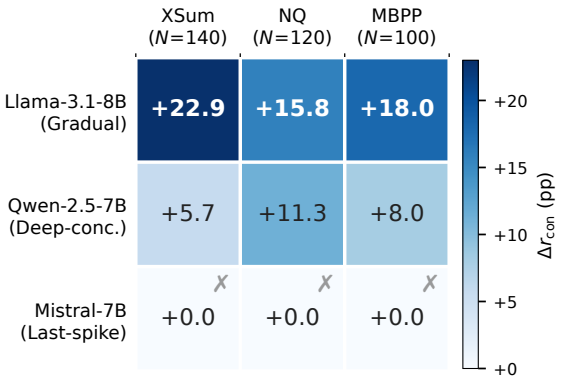

Figure 4: Consistency improvement (Δr_{con}, in pp) over Vanilla across three tasks (columns) and three judge models (rows). The row annotations indicate the bias-distribution type identified by probing (§4.2); cells marked with × exhibit no response to steering.

Source → Target	Δr_{con}	Retention
NQ → XSum	+19.3 pp	84%
MBPP → XSum	+18.6 pp	81%
MBPP → NQ	+15.8 pp	100%
XSum → NQ	+14.2 pp	89%
XSum → MBPP	+5.0 pp	28%
NQ → MBPP	+4.0 pp	22%

Table 1: Cross-task transfer on `Llama-3.1-8B`. Retention = $\Delta r_{\text{con}}^{\text{transfer}} / \Delta r_{\text{con}}^{\text{in-task}}$. All six directions are positive.

Task	Vanilla	Steered	Δr_{con}
NQ	.283	.717	+43.3 pp
MBPP	.280	.650	+37.0 pp
XSum	.050	.514	+46.4 pp

Table 2: Cross-model transfer: Llama-3.1-8B → Llama-3-8B (same-family).

structures across NLP tasks, though specialized domains exhibit distinct domain-specific shifts.

Cross-model transfer. A steering vector trained on Llama-3.1-8B-Instruct transfers to Llama-3-8B-Instruct with even larger gains (from +37 pp to +46 pp; Table 2), because the transferred vector exploits the stronger vanilla position bias of the older model (for example, an XSum consistency of 0.050 versus 0.500). This result suggests that positional representations are shared within the Llama family.

4.5 Layer Selection Ablation

To validate the max|bias|-layer selection effective, we train separate steering vectors at eight layers of Llama-3.1-8B on the XSum training set (Figure 5).

The max|bias| layer is the decisive winner. Layer 24, the max|bias| layer identified by probing, achieves the highest consistency (0.729) and lowest asymmetry (0.014). The deep layers (L20–L28) form a stable high-performance band ($r_{\text{con}} \geq 0.700$), whereas the shallow layers (L14 and L17) fall into a performance valley (0.664–0.671).

Shallow-layer instability. Layer 4 attains a surprisingly high accuracy (0.521) at $\alpha = 1.0$, but its operating window is exceedingly narrow: raising α to 1.5 collapses the consistency to 0.071. The steering signal traverses 28 subsequent layers before it reaches the output logits, so small perturbations are amplified unpredictably. This finding confirms that max|bias|-localized intervention is essential for robust deployment.

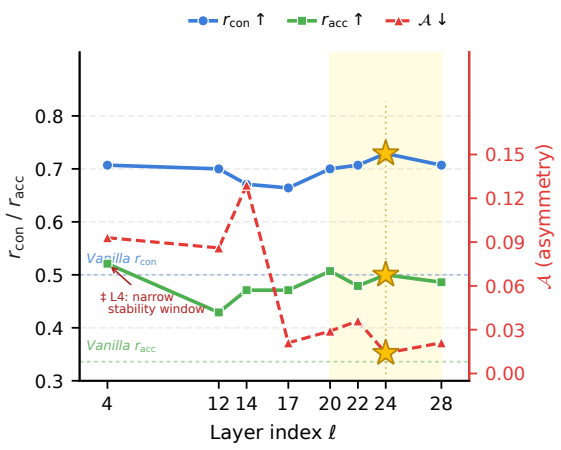

Figure 5: Layer ablation on Llama-3.1-8B-Instruct. Dotted horizontal lines mark the Vanilla baselines. The yellow band highlights the deep-band plateau (L20–L28, $r_{\text{con}} \geq 0.700$). The gold ★ marks L24, the max|bias| layer; ‡L4 attains a high accuracy but exhibits a narrow stability window (collapses at $\alpha \geq 1.5$). Numerical values for each cell are tabulated in Appendix G.

5 Discussion

5.1 Analysis of Failed Baselines

Attention Sorting, PCW, and PINE were originally designed for fundamentally different scenarios from pairwise evaluation and consequently fail despite reasonable adaptation. Attention Sorting (Peysakhovich and Lerer, 2023) degenerates into the vanilla baseline ($r_{\text{con}} = 0.500$, $r_{\text{acc}} = 0.336$): in the two-option setting, attention mass consistently concentrates on position 2, so the sort step never triggers a swap (swap rate $= 0$). PCW (Ratner et al., 2023) collapses to $r_{\text{con}} = 0.057$ with position-1 selection rate of 0.907: isolating each candidate in its own attention window removes the cross-option comparison signal that the judge needs to compare two summaries, and the model therefore defaults to position 1. PINE (Wang et al., 2025), whose importance-based position reallocation targets the ranking of five or more documents, similarly collapses to $r_{\text{con}} = 0.071$ with a position-1 rate of 0.921: in the two-option case, the reallocation degenerates into a near-fixed positional assignment that injects an even stronger bias than the original. These failures share a common cause. These methods are designed for settings in which multiple candidates are mutually independent and exhibit different degrees of relevance to the query, and they therefore do not transfer to the pairwise setting, in which the two candidates are equally task-relevant.

5.2 Method Applicability and Bias Distribution Patterns

The three bias-distribution patterns identified by probing (§4.2) delineate the applicability boundary of residual-stream steering. PaS is highly effective on *gradual-accumulation* models (Llama-3.1-8B): the bias spans the mid-to-deep layers, which provides a wide and forgiving intervention window. On *deep-concentrated* models (Qwen-2.5-7B), the effectiveness of PaS diminishes: the bias spikes sharply at L25–26 (out of 28), with only 2–3 layers of leverage distance to the unembedding, which restricts the achievable gains. PaS is ineffective on *last-layer spike* models (Mistral-7B): the bias is introduced almost entirely at L31, and residual-stream perturbations immediately preceding the unembedding produce ill-conditioned gradients that drive the steering vector to a trivial near-zero solution ($\|\mathbf{v}\|_2 \approx 0.006$). We conclude that residual-stream steering succeeds when position bias spans multiple mid-to-deep layers with sufficient "leverage distance" to the output logits, but breaks down when the bias is concentrated at the final layer. For the latter regime, logit-level calibration is likely a more appropriate alternative. We accordingly recommend that practitioners first run the probing diagnostic (§3.2). This diagnostic checks whether a judge lies within PaS's operating range.

6 Conclusion

We introduce Probe-and-Steer (PaS), the first probe-localized, single-layer steering approach for mitigating position bias in pairwise LLM-as-a-Judge. By treating position bias as a *local representational phenomenon* rather than a distributed pathology, our work shifts the design space of debiasing from prompt-, attention-, or channel-level calibration toward representation-level steering, and it does so through a parameter-light intervention that keeps the judge fully frozen. The probing-based applicability analysis further delineates when residual-stream steering is the right tool. Taken together, our work provides a principled, low-cost procedure for diagnosing and mitigating position bias.

Future Work. Two directions naturally extend this work: scaling probe-localized steering to larger models (70B+) to test whether the bias-critical layers shift predictably with depth, and developing alternative interventions for last-layer spike models (for example, logit-space calibration or multi-head output steering).

Limitations

- **Model-Dependent Efficacy.** PaS produces double-digit consistency gains on Llama-3.1-8B across all three tasks, but Mistral-7B-Instruct shows no response to steering on any task. Different model families may encode position bias in structurally distinct layers or subspaces that our method cannot always address.
- **Task Coverage.** We evaluate PaS on three tasks: summarization, open-domain QA, and code generation. Other LLM-as-a-Judge scenarios, such as mathematical reasoning and safety filtering, may require task-specific retraining of the steering vector. Although the cross-task transfer results are positive, they are limited to these three domains.
- **Resource Constraints.** Due to limited computational resources, our experiments are restricted to open-source judges in the 7–8B parameter range. Whether the bias-critical layer structure and the effectiveness of probe-localized steering generalize to substantially larger models (e.g., 70B-parameter judges) remains an open question that we leave to future work.

Ethics Statement

1. **Bias and Fairness.** Mitigating position bias in pairwise LLM-as-a-Judge improves the fairness and reliability of large-scale automated evaluation.
2. **Potential Misuse.** The steering-vector technique can equally be applied to *inject* bias rather than to mitigate it. We recommend that any steering method disclose the steering layer, the vector norm, and the training objective in the model card, in order to enable independent auditing and verification of the direction and magnitude of the intervention.

References

Dyah Adila, Shuai Zhang, Boran Han, and Bernie Wang. 2024. Discovering bias in latent space: An unsupervised debiasing approach. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 246–261. PMLR.

Anthropic. 2025. System card: Claude Sonnet 4.5.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

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 32 others. 2022. Constitutional AI: Harmlessness from AI feedback. *Preprint*, arXiv:2212.08073.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. UltraFeedback: Boosting language models with scaled AI feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 9722–9744. PMLR.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

DeepSeek-AI. 2024. DeepSeek-V3 technical report. *Preprint*, arXiv:2412.19437.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaFarm: A simulation framework for methods that learn from human feedback. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*.

Gemini Team, Google. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shahriar Golchin and Mihai Surdeanu. 2024. Time travel in LLMs: Tracing data contamination in large

language models. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*.

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. *Preprint*, arXiv:2407.21783.

Junqing He, Kunhao Pan, Xiaoqun Dong, Zhuoyang Song, Yibo Liu, Qianguo Sun, Yuxin Liang, Hao Wang, Enming Zhang, and Jiaxing Zhang. 2024. Never lost in the middle: Mastering long-context question answering with position-agnostic decompositional training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13628–13642, Bangkok, Thailand. Association for Computational Linguistics.

Jason Hoelscher-Obermaier, Julia Persson, Esben Kran, Ioannis Konstas, and Fazl Barez. 2023. Detecting edit failures in large language models: An improved specificity benchmark. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11548–11559, Toronto, Canada. Association for Computational Linguistics.

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

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 26874–26901. PMLR.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2025. From crowdsourced data to high-quality benchmarks: Arena-Hard and BenchBuilder pipeline. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 34209–34231. PMLR.

Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2024. Split and merge: Aligning position biases in LLM-based evaluators. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11084–11108, Miami, Florida, USA. Association for Computational Linguistics.

Sheng Liu, Haotian Ye, Lei Xing, and James Y. Zou. 2024. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32287–32307. PMLR.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

Alexander Peysakhovich and Adam Lerer. 2023. Attention sorting combats recency bias in long context language models. *Preprint*, arXiv:2310.01427.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6383–6402, Toronto, Canada. Association for Computational Linguistics.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2025. Judging the judges: A systematic study of position bias in LLM-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 292–314, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Asa Cooper Stickland, Alexander Lyzhov, Jacob Pfau, Salsabila Mahdi, and Samuel R. Bowman. 2024. Steering without side effects: Improving post-deployment control of language models. *Preprint*, arXiv:2406.15518. Presented at the NeurIPS 2024 Workshop on Safe Generative AI.

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

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *Preprint*, arXiv:2308.10248.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M. Kakade, Hao Peng, and Heng Ji. 2025. Eliminating position bias of language models: A mechanistic approach. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Wang, Bo Li, Bowen Liu, Chengyuan Lin, Dayiheng Lu, and 1 others. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Yijiong Yu, Huiqiang Jiang, Xufang Luo, Qianhui Wu, Chin-Yew Lin, Dongsheng Li, Yuqing Yang, Yongfeng Huang, and Lili Qiu. 2025. Mitigate position bias in LLMs via scaling a single hidden states channel. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6092–6111, Vienna, Austria. Association for Computational Linguistics.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023) Datasets and Benchmarks Track*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2025. JudgeLM: Fine-tuned large language models are scalable judges. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*.

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. 2023. Representation engineering: A top-down approach to AI transparency. *Preprint*, arXiv:2310.01405.

A Inference Cost Analysis

Table 3 compares the inference cost of PaS with that of all baselines on 140 XSum test samples on a single A100 GPU.

The marginal overhead of PaS consists of a single 4,096-dimensional vector addition at one layer per forward pass, which is negligible relative to the 8B-parameter forward computation. By contrast, prompt-level or attention-level interventions (Attention Sorting, PCW, and PINE) require disabling FlashAttention and resorting to standard attention recomputation, which inflates wall-clock latency by $+260\%$ to $+380\%$; SteerFair requires per-head activation extraction and projection at every layer, adding approximately 12% overhead. The single-layer design of PaS therefore makes it the most cost-efficient mitigation method in our comparison while simultaneously achieving the strongest performance.

Method	Wall-clock	Overhead
Vanilla	$\sim$25 min	—
Attention Sorting	$\sim$120 min	$+380\%$
PCW	$\sim$90 min	$+260\%$
PINE	$\sim$100 min	$+300\%$
SteerFair	$\sim$28 min	$+12\%$
Pos. Hidden Scaling	$\sim$27 min	$+8\%$
PaS (Ours)	**$\sim$26 min**	**$+4\%$**

Table 3: Inference cost comparison on 140 XSum test samples (single A100). Methods that need to disable FlashAttention (Dao et al., 2022) (AS, PCW, PINE) incur substantially higher wall-clock overhead due to recomputation in standard attention. PaS achieves the lowest overhead among all bias-mitigation methods.

B Single-Layer vs. Multi-Layer Ablation

A natural question is whether injecting steering vectors at multiple layers offers any improvement over a single-layer intervention. We test this hypothesis by training PaS with multi-layer configurations centered on L20 (Table 4).

Multi-layer steering degrades performance. The 5-layer configuration collapses entirely to vanilla-level performance (consistency = 0.500 and $\mathcal{A} = 0.471$), which effectively nullifies all steering. The 3-layer configuration (L19–21) exhibits intermediate degradation (-2.9 pp in consistency). Even widely spaced layers (L16, L20, and L24) fail in a similar manner, which confirms that the issue is not merely adjacent-layer interference.

Failure mechanism. A detailed per-α analysis reveals that Multi-5 at $\alpha \in [0.8, 1.2]$ *amplifies* rather than cancels the position-1 anchoring, with $\mathcal{A}$ rising well above the Vanilla baseline (0.471). This finding indicates that jointly optimized vectors can learn opposing directions that reinforce rather than cancel the bias. We attribute this behavior to gradient conflicts: when vectors at multiple layers are jointly optimized, the gradient of each vector is computed through the interventions of the other vectors, which creates a coupled optimization landscape in which corrections at one layer cancel or reverse corrections at others. The single-layer intervention avoids this issue entirely and remains the most parameter-efficient choice.

Config	Layers	$r_{\text{con}} \uparrow$	$r_{\text{acc}} \uparrow$	$\mathcal{A} \downarrow$
Single	[20]	**.700**	**.507**	**.029**
Multi-3	[19, 20, 21]	.671	.471	.057
Multi-5	[18–22]	.500	.336	.471
Multi-wide	[16, 20, 24]	.521	.350	.464

Table 4: Single-layer vs. multi-layer steering on XSum ($N = 140$). Multi-layer configurations train independent vectors at each layer and apply them simultaneously.

C Steering Strength α Robustness

We sweep $\alpha \in \{0, 0.1, 0.2, \ldots, 3.0\}$ over 19 points at layer 20 to characterize the sensitivity of PaS to the steering magnitude (Figure 6 and Table 5). We conduct the α-sweep at L20 rather than at the main-result layer L24 because L20 served as the default during our multi-task experiments. The layer ablation (§4.5) shows that L20 and L24 share the same qualitative α-response pattern within the deep-layer band.

Three regimes emerge. (i) *Ramp-up* ($\alpha \leq 0.5$): consistency rises monotonically from 0.500 to 0.607 and $\mathcal{A}$ drops from 0.471 to 0.293 as the vector begins to override the position-1 anchoring. (ii) *Plateau* ($\alpha \in [1.2,\ 1.4]$): consistency peaks at 0.700 (at $\alpha = 1.3$) and accuracy peaks at 0.507, while $\mathcal{A}$ remains within ± 0.05; three consecutive values yield consistency within ± 1 pp, which confirms the **robustness to** α. (iii) *Over-correction* ($\alpha \geq 1.5$): $\mathcal{A}$ rises sharply as the vector overshoots toward position 2, and consistency declines to 0.414 at $\alpha = 3.0$.

Practical recommendation. We default to $\alpha = 1.2$ at L24 (or to $\alpha = 1.3$ at L20). The $[1.2, 1.4]$ plateau implies that practitioners need no fine-grained tuning: any value of α in this range matches the optimum within ± 1 pp. The minimum of $\mathcal{A}$ (0.021 at $\alpha = 1.2$) coincides with this plateau,

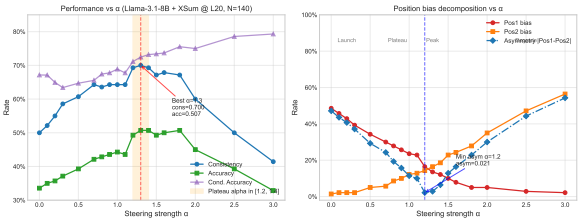

Figure 6: Steering strength sweep on layer 20 (19 points). **Left:** consistency and accuracy vs. α; the $[1.2, 1.4]$ plateau is shaded. **Right:** ASYMMETRY-INDEX $\mathcal{A}$ vs. α; $\mathcal{A}$ attains its minimum near $\alpha = 1.2$.

α	$r_{\text{con}} \uparrow$	$r_{\text{acc}} \uparrow$	$\mathcal{A} \downarrow$
0.0	.500	.336	.471
0.1	.521	.350	.436
0.2	.550	.357	.407
0.3	.586	.371	.371
0.5	.607	.393	.293
0.7	.643	.421	.243
0.8	.636	.429	.193
0.9	.643	.436	.157
1.0	.643	.443	.114
1.1	.643	.436	.100
1.2	**.693**	**.493**	**.021**
1.3	**.700**	**.507**	.029
1.4	**.693**	**.507**	.064
1.5	.671	.493	.129
1.6	.679	.500	.164
1.8	.671	.507	.229
2.0	.600	.450	.300
2.5	.500	.393	.443
3.0	.414	.329	.543

Table 5: Complete α-sweep at layer 20 on XSum ($N = 140$). The $[1.2, 1.4]$ plateau is bolded; $\alpha = 1.3$ achieves the highest consistency and accuracy.

which indicates that the steered judge approaches a near-symmetric residual bias at the recommended steering strength.

D Loss Function Ablation

To verify that the three objectives target orthogonal facets of position bias, we train PaS at layer 20 with each individual loss, with each pairwise combination, and with the full triple (Table 6). We use layer 20 for all loss and λ ablations in order to maintain direct comparability across configurations. The qualitative conclusions also transfer to L24, because both layers reside in the same deep-band performance plateau (§4.5).

Each loss plays a specialized role. (i) $\mathcal{L}_2$ (Consistency) alone recovers 99% of the consistency gain of the full model (0.707 versus 0.700) and achieves the lowest single-loss asymmetry (0.007); it is therefore the primary driver of position-invariant behavior. (ii) $\mathcal{L}_1 + \mathcal{L}_3$ achieves *perfect*

Loss	$r_{\text{con}} \uparrow$	$r_{\text{acc}} \uparrow$	$\mathcal{A} \downarrow$	α
$\mathcal{L}_1$ only	.636	.471	.279	0.8
$\mathcal{L}_2$ only	**.707**	.493	.007	1.5
$\mathcal{L}_3$ only	.686	.479	.057	1.5
$\mathcal{L}_1 + \mathcal{L}_2$	.686	**.514**	.129	1.0
$\mathcal{L}_1 + \mathcal{L}_3$	.686	.493	**.000**	1.2
$\mathcal{L}_2 + \mathcal{L}_3$	.679	.471	.050	1.5
Full ($\mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$)	.700	.507	.029	1.3

Table 6: Loss ablation at layer 20 ($N = 140$ test).

symmetry ($\mathcal{A} = 0.000$), which confirms that invariance and calibration jointly enforce balanced position treatment. (iii) $\mathcal{L}_1 + \mathcal{L}_2$ yields the highest accuracy (0.514), which shows that calibration is not the sole driver of accuracy.

The full combination provides a balanced trade-off. The full model does not achieve the best single-metric score on any axis, but it uniquely avoids weakness on any axis: it simultaneously attains a consistency of 0.700, an accuracy of 0.507, and a low asymmetry of 0.029. Each narrower configuration trades off one metric for another: $\mathcal{L}_1 + \mathcal{L}_2$ maximizes accuracy but inflates asymmetry to 0.129 (versus 0.029), and $\mathcal{L}_1 + \mathcal{L}_3$ attains perfect symmetry but at 1.4 pp lower accuracy. $\mathcal{L}_3$ supplies the only explicit content-grounded supervision in our objective and is therefore required in order to reach this balanced operating point.

E Loss Weight λ Sensitivity

Table 7 reports the sensitivity of PaS to the choice of loss weights. We vary one weight at a time while keeping the others fixed at their default values ($\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_3 = 2$, $\lambda_{\text{mag}} = 0.01$).

Robustness to λ_3. Varying λ_3 (the calibration weight) across an $8\times$ range from 0.5 to 4.0 changes consistency by at most 0.7 pp (0.693–0.700) and accuracy by at most 1.4 pp (0.500–0.514). PaS is therefore highly insensitive to this hyperparameter.

Effect of individual weights. Doubling λ_2 (the consistency weight) reduces asymmetry to 0.014, which is the lowest across all configurations and confirms that the consistency objective drives symmetric bias reduction. Doubling λ_1 (the invariance weight) slightly degrades all metrics (-2.1 pp in consistency), because an excessively strong uniformity pressure suppresses the ability of the model to discriminate content. The default configuration is near-optimal across all metrics simultaneously.

Config	λ_1	λ_2	λ_3	$r_{\text{con}}\uparrow$	$r_{\text{acc}}\uparrow$	$\mathcal{A}\downarrow$
$\lambda_3 = 0.5$	1	1	0.5	.693	.500	.021
$\lambda_3 = 1.0$	1	1	1.0	.693	.507	.193
Default	**1**	**1**	**2.0**	**.700**	.507	.029
$\lambda_3 = 4.0$	1	1	4.0	.700	**.514**	.043
$\lambda_1 = 2.0$	2	1	2.0	.679	.493	.050
$\lambda_2 = 2.0$	1	2	2.0	.700	.500	**.014**

Table 7: Loss weight (λ) sensitivity at layer 20 on XSum ($N = 140$). Best-α result for each configuration.

F Prompt Templates

We provide the full system prompts and user-message templates used for pairwise evaluation across all three tasks. These prompts are shared by PaS and all baselines, and only the intervention mechanism differs.

F.1 XSum (Summarization)

System prompt.

```
You are a helpful assistant and a news-article summarizer. You help compare summaries to help me with my records. You must respond with only "1" or "2" and no other text.
```

User message template.

```
Here are two news-article summaries I found. It's important for me to select the better summary for my records.

Article: {article}
Summary1: {summary1}
Summary2: {summary2}

Can you tell me which summary is better? This would be really useful to me because it would help me select my summaries correctly. Please answer with only "1" or "2" and no other text.
```

F.2 NQ (Open-Domain QA)

System prompt.

```
You are an impartial judge evaluating answers to open-ended questions. You compare two answers and decide which is more helpful, accurate, and relevant. You must respond with only "1" or "2" and no other text.
```

User message template.

```
Here are two answers to an open-ended question. It's important for me to select the better answer for my records.

Question: {question}
Answer1: {answer1}
Answer2: {answer2}

Can you tell me which answer is better? Consider helpfulness, accuracy, and relevance to the question. This would be really useful to me because it would help me select my answers correctly. Please answer with only "1" or "2" and no other text.
```

F.3 MBPP (Code Generation)

System prompt.

```
You are an impartial judge evaluating Python code solutions. You compare two solutions and decide which one is better in correctness, readability, and efficiency. You must respond with only "1" or "2" and no other text.
```

User message template.

```
Here are two Python code solutions to a programming problem. It's important for me to select the better solution for my records.

Problem: {problem}
Example tests: {tests}

Solution1: {code1}
Solution2: {code2}

Can you tell me which solution is better? Consider correctness against the example tests, readability, and efficiency. This would be really useful to me because it would help me select my solutions correctly. Please answer with only "1" or "2" and no other text.
```

Remarks. All prompts adopt a "replier-unaware" format: the judge is not informed of the source of either candidate. The judge selects a winner by emitting a single token ("1" or "2") through greedy decoding.

G Per-Layer Steering Detailed Ablation

Table 8 provides the complete per-α results for all eight layers tested in the layer ablation (§4.5). The steering vector for each layer is trained independently on the same 100 training samples and with the default loss weights, and it is then evaluated at seven steering strengths.

Stability versus depth. The per-α breakdown confirms the shallow-layer instability discussed in §4.5: the deep layers (L20–L28) maintain stable performance across all tested values of α, which provides practitioners with a wide and forgiving operating range.

Layer	$\alpha=0$	$\alpha=0.5$	$\alpha=0.8$	$\alpha=1.0$	$\alpha=1.2$	$\alpha=1.5$	$\alpha=2.0$
4	.50/.34/.47	.64/.44/.13	.69/.49/.07	**.71/.52/.09**	.68/.49/.15	.07/.05/.92	.31/.22/.50
12	.50/.34/.47	.59/.39/.28	.64/.43/.19	.66/.49/.17	.69/.43/.07	.66/.39/.06	**.70/.43/.09**
14	.50/.34/.47	.61/.40/.36	.64/.44/.30	.66/.46/.21	**.67/.47/.13**	.66/.48/.14	.53/.39/.47
17	.50/.34/.47	.59/.39/.28	.59/.41/.14	.61/.41/.06	.64/.43/.06	**.66/.47/.02**	.60/.44/.07
20	.50/.34/.47	.61/.39/.29	.64/.43/.19	.64/.44/.11	**.69/.49/.02**	.67/.49/.13	.60/.45/.30
22	.50/.34/.47	.63/.41/.26	.66/.44/.14	.69/.47/.06	**.71/.48/.04**	.66/.44/.04	.54/.39/.41
24	.50/.34/.47	.63/.41/.26	.67/.45/.14	.70/.48/.06	**.73/.50/.01**	.66/.45/.06	.51/.37/.46
28	.50/.34/.47	.63/.41/.23	.67/.45/.11	.70/.48/.03	**.71/.49/.02**	.66/.46/.03	.56/.39/.36

Table 8: Per-layer, per-α results on XSum ($N=140$). Each cell shows consistency / accuracy / asymmetry. Best α per layer is **bolded**.

H Terms of Use and Data Documentation

This section documents the licenses of all scientific artifacts used in this work and discusses data content considerations.

H.1 Licenses for Models

- **Llama-3.1-8B-Instruct** and **Llama-3-8B-Instruct** (Grattafiori et al., 2024): Meta Llama 3.1 Community License Agreement (https://llama.meta.com/llama3_1/license/).
- **Qwen-2.5-7B-Instruct** (Yang et al., 2024): Apache License 2.0 (https://choosealicense.com/licenses/apache-2.0/).
- **Mistral-7B-Instruct-v0.3** (Jiang et al., 2023): Apache License 2.0.
- **GPT-3.5-Turbo** (Ouyang et al., 2022): OpenAI Terms of Use (https://openai.com/policies/terms-of-use).
- **Gemini 2.5 Pro** (Gemini Team, Google, 2025): Google API Terms of Service (https://ai.google.dev/gemini-api/terms).
- **DeepSeek-V3** (DeepSeek-AI, 2024): DeepSeek License (https://github.com/deepseek-ai/DeepSeek-V3/blob/main/LICENSE-MODEL).
- **Claude Sonnet 4.5** (Anthropic, 2025): Anthropic Terms of Service (https://www.anthropic.com/legal/consumer-terms).

H.2 Licenses for Datasets

- **XSum** (Narayan et al., 2018): MIT License (https://github.com/EdinburghNLP/XSum?tab=MIT-1-ov-file).
- **Natural Questions** (Kwiatkowski et al., 2019): CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0/).
- **MBPP** (Austin et al., 2021): CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

All artifacts are used solely for non-commercial research purposes, consistent with their respective licenses and intended use.

H.3 Personal Information and Offensive Content

The datasets used in this study are established public benchmarks that have been curated by their original creators. XSum articles are derived from BBC news reports and may contain named entities of public figures and locations, but consist entirely of publicly available news text without private personal information. Natural Questions are derived from anonymized Google search queries paired with Wikipedia passages and contain no private personal identifiers. MBPP consists of synthetic programming problems with no personal or offensive content. Our constructed evaluation pairs are derived exclusively from these source datasets using model-generated candidate responses and introduce no additional personal information or offensive content.