BIAS DISCOVERY AND STEERING IN VLM JUDGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) serve as multimodal judges in evaluation, training, and inference, but they exhibit biases that propagate to each of these settings. Existing methods for detecting biases typically analyze judge outputs, either testing for specific biases or discovering them through properties of the evaluated responses. Those that do examine the judge's internal states focus on predefined biases. We instead discover biases from these states without choosing them in advance, and correct them by modifying the same states. Using sparse autoencoders, we extract interpretable features from hidden states and introduce a bias score that measures how differently the judge and humans weigh each feature. We then steer the judge along selected features, with a lightweight policy adapting the intervention to each input. Applied to two judge models and two datasets, our method surfaces biases spanning safety, visual grounding, instruction following, and multilingual interaction, including several that are largely unexplored in judges. Steering with bias features produces statistically significant improvements in accuracy for both judges, with increases on relevant task subsets of up to 11.6 percentage points.

1 INTRODUCTION

Large language models (LLMs) are used as judges that compare the quality of model-generated responses, offering a scalable substitute for human evaluation (Zheng et al., 2023). Vision-language models (VLMs) fill the same role for multimodal tasks (Chen et al., 2024). However, they inherit biases found in LLM judges, including unjustified preferences for certain positions (Wang et al., 2024) or longer responses (Hu et al., 2025), and add their own, such as overlooking visual evidence (Zou et al., 2026). These biases affect every use case of VLM judges, from labeling preference data for training (Li et al., 2024b) and ranking models on benchmarks (Lu et al., 2024) to selecting among sampled outputs at inference time (Xiong et al., 2025). Reducing the biases is therefore crucial.

Targeted mitigation requires first identifying the biases that the judge exhibits. Many studies test for biases hypothesized in advance (Koo et al., 2024; Ye et al., 2025) and cannot detect unanticipated ones. Several discovery methods address this by inferring preferences from data (Findeis et al., 2025; Movva et al., 2026; Wedgwood et al., 2026). They identify properties of the responses that predict which one is chosen, but do not examine the internal states behind the judge's verdicts. Their findings therefore cannot directly guide interventions on these states. Other work does act on them through steering, but only for biases selected beforehand (Sun et al., 2026; Xu et al., 2026).

To close these gaps, we propose a method that uses the judge's internal states to both discover and mitigate its biases (Figure 1). We decompose the hidden states preceding the verdict with sparse autoencoders (SAEs) (Bricken et al., 2023; Huben et al., 2024) to produce interpretable features. These represent patterns in the judge's comparative reasoning. To relate the features to biases, we introduce a *bias score* that applies to any feature and is computed using existing human preference data. It is the correlation between the feature's activation and the part of human preference that is left unexplained by the judge's preference; its sign and magnitude indicate whether the judge overvalues or undervalues the feature compared with humans, and by how much. Because the features are directions in the judge's activation space, steering along them can directly target the discovered biases. We present an adaptive policy that decides how to steer the judge for each input. The intervention is lightweight: training the policy requires no new judgments, and at inference, the steering is determined once from the prompt's hidden states before generation begins.

We apply our method to two VLM judges, Gemma 4 E4B (Gemma Team, 2026) and Qwen3.5-4B (Qwen Team, 2026), using human preference data from MM-RLHF (Zhang et al., 2025) and

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.

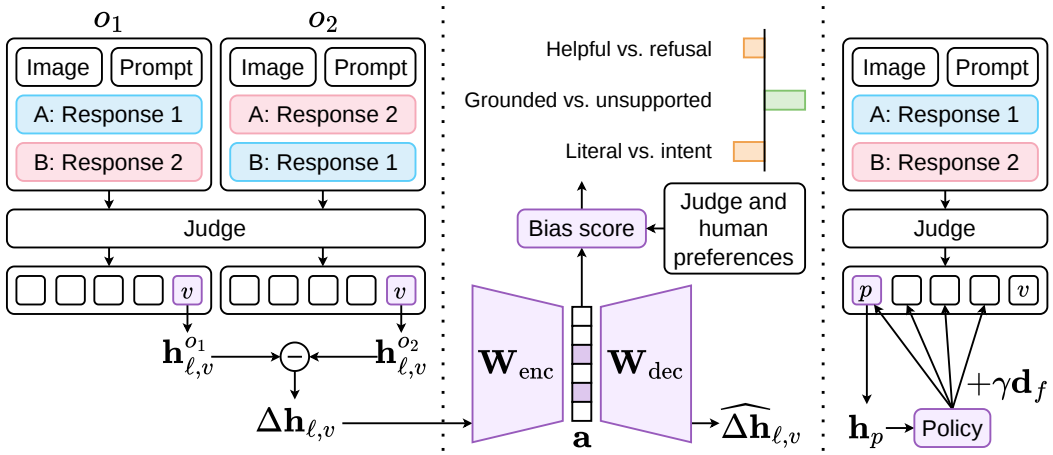

Figure 1: Overview of our method. **Left:** We compute the difference between the judge's hidden states at the verdict position v for the two orderings of each response pair. **Center:** An SAE trained on the differences decomposes them into features. The bias score of each feature indicates whether the judge places too much or too little weight on it relative to humans. **Right:** We steer the judge along a feature direction $\mathbf{d}_f$ during its thinking, with a coefficient γ that a trained policy chooses based on the hidden states at the last prompt position p.

VisionArena-Battle (Chou et al., 2025). The bias scores reveal features that capture a range of misaligned preferences, including a stronger aversion than humans to refusals of potentially harmful requests and a weaker preference for reasoning grounded in the image. We also recover the familiar length bias dimension and find other biases that are rarely studied in judges, such as prioritizing literal instruction following and undervaluing responses in the user's language. Steering along discovered features influences the verdict by reshaping the judge's reasoning and achieves statistically significant accuracy improvements for both models. The effects concentrate on the cases that each feature targets: for refusal, performance on relevant safety tasks rises by 11.6 percentage points for Gemma 4 and 11.5 for Qwen3.5.

2 Related Work

Biases in LLM and VLM judges. LLM judges exhibit various biases, including unwarranted preferences based on position or length (Zheng et al., 2023; Wang et al., 2024; Hu et al., 2025). To quantify these tendencies, prior work designs tests tailored to predefined bias types (Koo et al., 2024; Ye et al., 2025). Similar tests reveal distinct biases in how VLM judges integrate and prioritize visual and textual evidence (Lee et al., 2026; Zou et al., 2026). Bias can also be understood as a mismatch between human and judge preferences, which Li et al. (2024a) examine by comparing how the two weigh a fixed set of response properties. Our bias score extends this perspective to arbitrary features, directly measuring the direction and strength of the mismatch.

Preference discovery. Learning the preferences of humans and judges directly from data can uncover previously unknown biases. Findeis et al. (2025) prompt an LLM to derive natural-language principles that explain pairwise preference data. Movva et al. (2026) train an SAE on differences between response embeddings, then identify which of its features predict annotator choices. Applying this approach to LLM judges, Wedgwood et al. (2026) identify biases as features that judges and humans favor differently. While these methods describe preferences in terms of the responses being evaluated, we analyze the judge's internal representations, discovering biases in how it reasons about the responses rather than only in what it prefers.

Activation steering. Activation steering changes model behavior at inference time by adding a vector to its hidden states (Turner et al., 2024; Rimsky et al., 2024). Some approaches adapt the intervention to each input (Lee et al., 2025; Parekh et al., 2025). Existing methods for bias correction

steer along directions for specified biases (Adila et al., 2024; Sun et al., 2026; Xu et al., 2026). We instead steer with bias features identified by our method, connecting discovery to mitigation.

3 METHOD

3.1 HIDDEN STATE COLLECTION

We examine a VLM acting as a pairwise judge. For each task, the judge receives an image, a text prompt containing a question or request related to the image, and a pair of candidate responses to the prompt, labeled as produced by AI assistants A and B. It is instructed to compare the two responses inside its thinking block and output a verdict "A" or "B" indicating the assistant whose response it considers to be better. The complete judgment generation prompt is in Appendix F.1.

The hidden states of the judge immediately before the verdict token carry the information from the reasoning that underlies the verdict. We therefore collect the residual stream activations $\mathbf{h}_{\ell,v}$ after each transformer layer ℓ at this verdict position v for each task.[1] To isolate reasoning patterns relevant to the comparative judgment of the responses, we propose to use the judge's hidden states from the two orderings of each response pair, o_1 and o_2, which swap the responses between position A, shown first, and position B, shown second. We generate a separate judgment for each ordering and compute a hidden state difference:

$$\Delta\mathbf{h}_{\ell,v} = \mathbf{h}^{o_1}_{\ell,v} - \mathbf{h}^{o_2}_{\ell,v}. \tag{1}$$

We assign o_1 and o_2 uniformly at random for each response pair. The intent of the hidden state subtraction is to remove shared task information while preserving the judge's relative assessment of the pair to facilitate the extraction of comparative reasoning features.

3.2 FEATURE DECOMPOSITION

The hidden state difference is a single combined representation of the judge's comparison. To decompose it into interpretable features, we use sparse autoencoders (SAEs) (Bricken et al., 2023; Huben et al., 2024). For each layer ℓ, we train an SAE to reconstruct $\Delta\mathbf{h}_{\ell,v}$:

$$\mathbf{a} = \mathrm{AbsBatchTopK}\big(\mathbf{W}_{\mathrm{enc}}\Delta\mathbf{h}_{\ell,v}\big), \qquad \widehat{\Delta\mathbf{h}}_{\ell,v} = \mathbf{W}_{\mathrm{dec}}\mathbf{a}. \tag{2}$$

The encoder weight matrix $\mathbf{W}_{\mathrm{enc}}$ maps the hidden state difference to feature pre-activations, which the AbsBatchTopK function sparsifies to produce the feature activation vector $\mathbf{a}$. The decoder then produces the reconstruction $\widehat{\Delta\mathbf{h}}_{\ell,v}$ as a weighted sum of the columns of its weight matrix $\mathbf{W}_{\mathrm{dec}}$, with the activations as the coefficients. Each decoder column is a learned unit-length feature direction.

AbsBatchTopK combines the BatchTopK (Bussmann et al., 2024) and AbsTopK (Zhu et al., 2026) activations as in Movva et al. (2026): during training, it retains top pre-activations across the batch by absolute value while preserving the signs, such that k values are selected on average per example. The remaining activations are set to zero. This absolute-value selection enforces sparsity while allowing the activation of a single feature to meaningfully span both positive and negative values, matching our symmetric comparative setting. At inference time, we replace the selection rule with a fixed magnitude threshold targeting the same average sparsity. Further details are in Appendix A.1.

3.3 BIAS SCORE

We introduce a metric that quantifies the strength and direction of bias corresponding to each learned SAE feature. Let z^o_A and z^o_B be the output logits of the judge for the verdict tokens "A" and "B", respectively, for a response pair presented in ordering o. Following the subtraction order in the hidden state difference, the continuous judge preference x combines these logits from both orderings to represent how strongly the judge favors the response assigned to A in o_1 over the other response. The human preference label y is $+1$ if the human-preferred response is in position A in o_1, and -1 otherwise. We fit a least-squares linear regression of y on x through the origin to obtain the slope β:

$$x = \big(z^{o_1}_A - z^{o_1}_B\big) - \big(z^{o_2}_A - z^{o_2}_B\big), \qquad \beta = \frac{\sum_i x_i y_i}{\sum_i x_i^2}, \tag{3}$$

[1] For the last layer, we use the normalized residual stream after the final RMSNorm of the model.

with response pairs indexed by i. Then the residual $y - \beta x$ is the part of the human preference that is unexplained by this linear prediction from the judge preference. We define the bias score for feature f as the Pearson correlation between its activation a_f and the residual:

$$\text{bias}_f = \text{corr}(a_f, y - \beta x) = \frac{r_{a_f,y} - r_{a_f,x}\, r_{x,y}}{\sqrt{1 - r_{x,y}^2}}, \tag{4}$$

where $r_{u,v}$ abbreviates $\text{corr}(u, v)$. All correlations are computed across response pairs and use population means of zero guaranteed by the random ordering.

Each SAE feature axis has an arbitrary sign; for bias scores, we choose the positive direction so that the activation correlates positively with judge preference. Then the sign of the bias score consistently indicates the direction of a mismatch in how the judge preference is associated with a feature relative to human preference. A negative score implies that the judge preference aligns with the feature more strongly than the feature's relationship with human preference would warrant under the linear fit. A positive score reflects the opposite mismatch: the relationship between the judge preference and the feature is weaker than expected relative to the human preference. The magnitude of the score measures the strength of the bias. This interpretation suggests reducing the influence of features with negative scores and increasing that of features with positive scores to improve agreement with human preference.

3.4 Feature interpretation

Qualitative interpretation characterizes the reasoning pattern that each feature represents beyond what its bias score conveys. This can be done by inspecting examples with the largest absolute activations. Each example consists of an image, prompt, candidate responses, and the judgments for both response orderings. To reduce the manual effort involved, we adapt automated interpretation procedures from prior work (Bills et al., 2023; Movva et al., 2026).

For each feature, we provide the top examples to a VLM and prompt it to identify a recurring comparative reasoning pattern in the judge's evaluations. To maintain semantic consistency, we present o_1 as the reference ordering for examples with positive activations and o_2 for those with negative activations. The model is instructed to describe the pattern from the perspective of the reference ordering, with the description applying in reverse for the other ordering. See Appendix F.2 for the prompt.

We validate each generated description using held-out examples with the largest absolute activations and zero activation. For each example, an LLM receives the description and the two judgments corresponding to the two response orderings in a random order. The model must select a judgment only if exactly one matches the description, and otherwise output "neither". Appendix F.3 shows the prompt. The ground-truth label is the reference-ordering judgment for examples with large activations and "neither" for zero-activation examples. We measure directional fidelity as the Pearson correlation between LLM predictions and ground-truth labels across active examples, encoding the two judgments as $+1$ and -1 and "neither" as 0. We also assess whether a description distinguishes the two judgments more often for active examples than for inactive examples. We quantify this selectivity by subtracting the proportion of non-"neither" predictions in inactive examples from that in active examples.

3.5 Feature steering

We use the discovered features to steer the judge and improve its alignment with human preference. During inference, we add a feature direction to the residual stream at its layer ℓ for token position t:

$$\mathbf{h}_{\ell,t} \leftarrow \mathbf{h}_{\ell,t} + \gamma \mathbf{d}_f, \tag{5}$$

where $\mathbf{d}_f$ is the SAE decoder column corresponding to the feature and γ is a signed steering coefficient that is fixed for each judgment. We restrict t to the positions immediately preceding each thinking token and stop steering before the generation of the verdict token so that the verdict is influenced only through the steered reasoning.

Steering policy. A signed feature direction represents a reasoning pattern tied to the A and B positions: how the judge perceives response A in contrast to response B. Steering along it requires deciding whether to steer at all, which side of the contrast to favor, and which position holds that side. We propose a steering policy that uses a learned score s to make the first two decisions, which depend on the task, and the feature's encoder pre-activation e to locate the two sides, which depends on the order of the responses. Both come from the judge's hidden states after it processes the inputs, and γ combines them:

$$s = \mathrm{scorer}_f\left(\frac{\mathbf{h}_p^o + \mathbf{h}_p^{o'}}{2}\right), \qquad e = \left[\mathbf{W}_{\mathrm{enc}}(\mathbf{h}_{\ell,p}^o - \mathbf{h}_{\ell,p}^{o'})\right]_f, \tag{6}$$
$$\gamma = \mathbf{1}\{|s| \geq \tau_f\} \cdot \mathrm{sign}(s) \cdot \mathrm{sign}(e) \cdot \alpha_f.$$

Scorer. The score s is produced by the feature-specific scorer_f, a multilayer perceptron (MLP) with a single hidden layer. For the scorer inputs, we define $\mathbf{h}_{\ell,p}$ as the layer ℓ hidden states at the last prompt token position p (the final token of the input to the judge), analogous to the verdict position hidden states $\mathbf{h}_{\ell,v}$ in Section 3.1. The superscript o indicates the ordering of the response pair as originally provided, and o' indicates the ordering with the responses swapped. These hidden states contain the judge's initial internal representation of the task and are cheaply available before generation. We concatenate $\mathbf{h}_{\ell,p}$ across all layers to form $\mathbf{h}_p$, then use its mean across the two response orderings as the input to scorer_f. Averaging the hidden states removes direct information about response positions and discourages the scorer from becoming another preference model. Our goal is instead to guide the reasoning using patterns that are useful or harmful for a given task type. This also keeps the score invariant to response ordering, consistent with its decisions depending only on the task content.

Direction and magnitude. The first factor of γ converts the score s into a decision on whether to steer. It is nonzero only when the magnitude of s meets the fixed per-feature threshold τ_f selected during policy training. The second factor $\mathrm{sign}(s)$ selects the side of the feature's contrast that the intervention is intended to favor. For example, suppose a feature represents the judge's focus on assistant A's detailed response compared to assistant B's brief response. Then for a response pair exhibiting this contrast, $\mathrm{sign}(s)$ indicates whether steering should favor the more detailed or briefer response, regardless of which position it appears in.

To translate this decision to the sign of γ, we must determine how the two response properties map to the A and B positions. We infer this assignment using the third factor $\mathrm{sign}(e)$. Here e is the pre-activation of feature f at the prompt-end position p, obtained by applying the SAE encoder to the hidden state difference at layer ℓ. Although the SAE was trained at the verdict position, we use the encoder here as an approximate feature detector in the judge's initial understanding. In the previous example, we interpret $e > 0$ as response A being more detailed and $e < 0$ as the reverse. The final factor α_f is a signed steering strength that is fixed per feature and sets the scale of the update.

Policy training. We train each scorer using proxy steering effects derived from the unsteered judge verdicts collected in Section 3.1 and their corresponding human preference labels. This avoids expensive generation of steered judgments. We train the scorer to output a positive s for the choice to prefer response A when $e > 0$ and B when $e < 0$, and a negative s for the opposite choice. Each sign of s selects for the same response property across orderings.

We define g_+ and g_- as the proxy accuracy gains from steering with positive and negative s, respectively. For each response pair, we assign $1 - \frac{c}{2}$ to the gain for the sign of s that is intended to favor the human-preferred response, and $-\frac{c}{2}$ to the other gain. Here $c \in \{0, 1, 2\}$ is the number of response orderings where the unsteered judge agrees with human preference. The proxy assumes that steering toward the human-preferred response corrects all incorrect verdicts and preserves correct ones, whereas steering toward the other response makes all correct verdicts incorrect and preserves incorrect ones.

To allow useful gradients, we replace the hard steering decision with a probability q_{steer}, obtained by dividing the threshold margin $|s| - \tau_f$ by a temperature T_{steer} and applying a sigmoid. The same transformation on s with temperature T_+ produces q_+, the probability of selecting the positive sign. We use the negative expected proxy gain as the loss:

$$\mathcal{L}_f = -\mathbb{E}\Big[q_{\mathrm{steer}}\big(q_+ g_+ + (1 - q_+) g_-\big)\Big]. \tag{7}$$

Table 1: Feature decomposition metrics for Gemma 4 E4B on MM-RLHF.

Method	Fidelity ↑	Significant features ↑	Selectivity (pp) ↑	Reconstruction error ↓
PCA	0.144	90	1.72	**0.161**
ICA	0.226	280	5.38	**0.161**
SAE	**0.249**	**325**	**7.18**	0.262

The expectation averages over response pairs in each training batch. We select τ_f at the start of training and after each epoch to maximize the mean training-set proxy gain under the hard steering rule. Appendix A.2 contains additional details.

4 Experiments

4.1 Setup

Datasets. We select two datasets to cover a broad range of tasks and preferences. MM-RLHF (Zhang et al., 2025) is a multimodal human-preference dataset spanning various topics including mathematics, captioning, commonsense reasoning, and safety. It contains model-generated responses ranked by humans based on quality. We use the image-based subset of its public release and filter for data quality, language, and response length to construct 39,413 preference pairs.

VisionArena-Battle (Chou et al., 2025) contains conversations from Chatbot Arena (Chiang et al., 2024) where users submitted preference votes for pairs of anonymous VLMs. The categories include captioning, entity recognition, homework, humor, and creative writing. After filtering the single-turn conversations for data quality and response length, we retain 13,462 preference pairs. Additional details on dataset preprocessing are in Appendix B.1.

Judge models. We use judge models from two model families with different architectures. Gemma 4 E4B (Gemma Team, 2026) has 42 layers and interleaves sliding window attention with full attention. Qwen3.5-4B (Qwen Team, 2026) has 32 layers comprising Gated DeltaNet (Yang et al., 2025) and gated attention (Qiu et al., 2025) layers. We run both models in thinking mode.

Hidden state collection. Our main analysis uses MM-RLHF with Gemma 4. We additionally apply our method to VisionArena-Battle with Gemma 4 and to MM-RLHF with Qwen3.5 to evaluate generalization across datasets and models, respectively. For each model–dataset combination, we generate judgments using the settings in Appendix B.2 and collect the hidden states. We keep preference pairs with correctly formatted judgments in both response orderings and randomly divide them into training, validation, and test sets in an 80:10:10 ratio. All pairs sharing an image are assigned to the same set.

4.2 Feature interpretability

Because bias discovery depends on interpretable features, we first assess whether SAEs produce such features from hidden state differences. For Gemma 4 on MM-RLHF, we train an SAE for each layer (Appendix B.3). Following previous work (Movva et al., 2026; Wedgwood et al., 2026), we set the number of features per SAE to 32 and use $k = 4$. To check whether simpler alternatives suffice, we fit principal component analysis (PCA) (Jolliffe, 2002) and independent component analysis (ICA) (Hyvärinen & Oja, 2000) to the same inputs with 32 components per layer.

We generate and validate feature descriptions for the three decompositions using the procedure and metrics described in Section 3.4. For PCA and ICA, whose activations are generally nonzero, we adapt the validation procedure to select the examples with the smallest absolute activations rather than those with zero activation. We use GPT-5.6 Luna with medium reasoning effort for both stages, with ten training examples per feature for description generation and 50 test examples each in the high-activation and low-activation groups for validation. Table 1 reports the results.

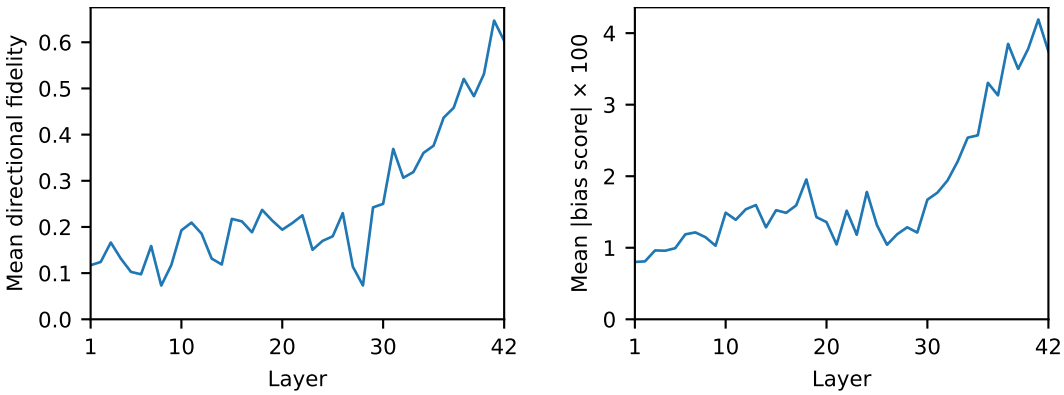

Figure 2: Mean directional fidelity (left) and mean absolute bias score (right) across layers for Gemma 4 E4B's SAE features on MM-RLHF.

Directional fidelity and selectivity are averaged over all 1,344 features, with zero assigned to values missing due to insufficient examples. We test for statistical significance of positive directional fidelity using an exact one-sided permutation test, with Bonferroni correction applied at a 0.05 level across all features for each method. The mean squared reconstruction error is evaluated on the test set, normalized by input variance and averaged over layers.

SAE features are more interpretable than PCA and ICA components. SAE outperforms both baselines on all three interpretability measures. Excluding the 47 SAE features with missing values raises its fidelity to 0.258 and selectivity to 7.44 percentage points, making the zero assignment conservative in this comparison. SAE trades reconstruction accuracy for greater interpretability while targeting only four active features per example, whereas PCA and ICA use all 32 components. These results support using SAE features for our subsequent analyses.

Interpretability and bias strength increase with depth. Figure 2 displays trends for directional fidelity and bias score magnitude for SAE features across Gemma 4's 42 layers. Bias scores are computed using data from the pooled training and validation sets. Both metrics rise steeply around layer 30, indicating that features in late layers tend to be more interpretable while representing stronger biases compared to early-layer features. At the verdict position where we collect the SAE's input hidden states, this late-layer region could be where the judge crystallizes its comparative considerations in preparation for the verdict.

4.3 Bias discovery

Next, we combine feature interpretations with bias scores to discover judge reasoning biases. We provide a selected list of distinctive bias features for Gemma 4 on MM-RLHF and VisionArena-Battle in Table 2. The features come from SAEs that are separately trained for each dataset and model layer, with 32 features per layer. The descriptions are shortened and edited from the LLM-generated interpretations in Appendix C, based on examination of top-activating examples. We place the side of the contrast that the judge tends to favor first, following the convention used to compute the bias scores.

Bias scores identify misaligned judge preferences. The first feature, "Helpful information vs. minimal refusal", activates strongly on tasks where one candidate response engages with the user's request and the other refuses based on potential for harm. Examples of such tasks include generating discriminatory captions or identifying private personal information based on an image. As "Helpful information" is presented first in the description according to our convention, we can infer that the judge has an overall relative preference for engagement over refusal. The negative bias score indicates that the judge places an overly large weight on helpfulness compared to the human annotators. Humans may also favor meaningful assistance at times, but they do so to a lesser degree than the judge on these tasks involving safety concerns.

Table 2: A selection of discovered bias features for Gemma 4 E4B.

Dataset	Layer	Feature description	Bias score × 100
MM-RLHF	17	Helpful information vs. minimal refusal	−2.93
	18	Contextual interpretation vs. listing isolated details	+2.45
	24	Image-grounded reasoning vs. unsupported answer	+5.74
	26	Detailed reasoning vs. minimal answers	+2.86
	29	Honest uncertainty vs. confident task completion	−1.67
	39	Negative conclusion vs. affirmative claim	−2.69
	40	Literal instruction adherence vs. intent fulfillment	−4.35
VisionArena-Battle	28	Task coverage vs. communicative impact	−2.20
	36	Substantive detail vs. matching the user's language	−1.00
	37	Attempt to answer vs. cautious refusal	+3.83

Layer 24's image grounding feature has a positive bias score and demonstrates misalignment in the opposite direction: the judge undervalues reasoning supported by visual evidence relative to human preference. Its bias score has the largest magnitude among the selected features, revealing a substantial discrepancy in the weight given to a fundamental criterion of multimodal evaluation.

Our method recovers a known bias dimension and discovers new ones. Length bias is a well-studied bias where the judge overvalues elaboration over brevity (Zheng et al., 2023; Hu et al., 2025). We find the same contrast in a layer 26 feature, but its positive bias score shows that the usual tendency is reversed. When the feature is active, Gemma 4 is more inclined than humans to choose a minimal response over a detailed explanation. Many of the other features capture biases that are less commonly discussed. For example, a layer 39 feature represents a bias toward "no" responses to yes-no questions, and a feature in layer 40 points to an excessive preference for literal compliance over fulfilling the user's intent.

Different datasets reveal different biases. Features for VisionArena-Battle correspond to biases that reflect the distinctive properties of the dataset. In its many creative writing tasks, users value vivid and engaging responses, but the judge overemphasizes addressing all parts of the request instead. The dataset's multilingual nature provides another source of disagreement: users prefer answers in their own language, while the judge puts more weight on the quality of the content.

VisionArena-Battle also has a feature related to refusal, but users favor engagement over caution more strongly than the judge does, reversing the MM-RLHF bias direction despite the model staying constant. This illustrates the need for selecting human preference data that is consistent with the intended alignment goals.

4.4 Bias steering

We demonstrate that steering with the discovered bias features can target the misaligned preferences and improve judgment accuracy. The steering effects of SAE features are known to be difficult to predict without empirical testing, with features varying in their ability to produce intended changes in model behavior (Arad et al., 2025). Our preliminary experiments show the same pattern and guide our choice of features for case studies of effective bias steering. For Gemma 4 on MM-RLHF, we use three features from Table 2: refusal (layer 17) for its relevance to safety, image grounding (layer 24) for its importance in judging responses to visual tasks, and length (layer 26) as an established source of judge bias. To evaluate steering across model architectures, we also identify features representing the same bias types in SAEs trained on Qwen3.5's hidden states.

We train a steering policy for each feature and select steering strengths using results on the validation set. Training parameters and steering settings are in Appendix B.4, and we provide an example of strength selection in Appendix D. Table 3 compares steered judge performance with unsteered baselines on the held-out test set with matching generation settings (Appendix B.2). Accuracy measures agreement with human preference, with judgments deviating from the specified output format counted as incorrect.

Table 3: Bias steering results for Gemma 4 E4B and Qwen3.5-4B on MM-RLHF.

Model	Bias feature	Accuracy (%)	Δ Accuracy (pp)		
			All [95% CI]	Subset	Other
Gemma 4	None (baseline)	77.6	—	—	—
	Refusal	79.2	**+1.7** [0.9, 2.5]	**+11.6**	+0.9
	Image grounding	78.5	**+1.0** [0.4, 1.6]	**+2.0**	+0.7
	Length	79.0	**+1.4** [0.5, 2.3]	**+2.1**	−0.5
Qwen3.5	None (baseline)	80.4	—	—	—
	Refusal	81.2	**+0.9** [0.4, 1.5]	**+11.5**	+0.1
	Image grounding	81.4	**+1.0** [0.4, 1.6]	**+2.2**	+0.7
	Length	81.4	**+1.0** [0.1, 1.9]	**+3.0**	−5.9

Bias steering improves agreement with human preference. The unsteered baselines achieve accuracies of 77.6% for Gemma 4 and 80.4% for Qwen3.5. Steering adds 0.9–1.7 percentage points (pp). We compute 95% confidence intervals (CIs) using a paired bootstrap procedure that resamples images, keeping associated judgments together. All six intervals lie above zero, indicating statistically significant improvement across both models and the three discovered bias types.

Benefits concentrate in relevant task subsets. To examine whether steering effects align with the feature interpretations, we define task subsets that are relevant to each bias. The refusal subset consists of tasks from MM-RLHF's safety category where humans rate one of the responses as more ethical than the other in the fine-grained annotations. We observe that these tasks often exhibit our target contrast between a refusal and an attempt to fulfill the request. The refusal bias features improve accuracy on this subset by 11.6 pp for Gemma 4 and 11.5 pp for Qwen3.5, substantially more than on the remaining tasks ("Other" in the table).

Similarly, we define the image grounding subset as tasks where only the human-preferred response contains the word "image", and the length subset as those where humans prefer the longer response. The image grounding and length bias features also produce larger gains within their respective subsets than on other tasks. Outside the subsets, the refusal and image grounding features increase accuracy, while the length feature introduces losses indicative of favoring the longer response even when humans prefer the shorter one.

Steering changes the reasoning and not just the verdicts. We inspect unsteered and steered judgments to understand how the intervention affects the reasoning behind the verdict. In one example, Gemma 4 evaluates responses to a request to explain an offensive meme. The unsteered judge recognizes the refusal as ethically responsible but selects the other response for fulfilling the request, explicitly distinguishing "better" from "safer". When steered with the refusal feature, the judge gives safety greater weight, prioritizing a "responsible, ethical, and helpful response" and choosing the human-preferred refusal. The generated text remains a coherent comparison of the two responses, with the revised verdict following naturally from the shift in emphasis. Appendix E contains the user prompt and complete judgment texts for the example.

5 Conclusion

We present a method for discovering and mitigating biases in a VLM judge: we decompose its activations into interpretable features, assign each feature a bias score that measures divergence from human preferences, and steer the judge along those same features. The features reveal both familiar and unfamiliar biases, and steering with a subset of them brings the judge's verdicts closer to human choices, especially where each bias matters most. Looking inside the judge allows us to connect the detection of biases directly to their correction. Directions for future work include a systematic way of finding features that are effective for steering, intervening on multiple biases simultaneously, and training that transfers the improvements into the judge's own weights, for example by learning from its steered counterpart.

AI USE STATEMENT

In this work, we used generative AI tools interactively and closely supervised their output, without delegating any task entirely to them. They assisted in refining the methodology, designing experiments, implementing the method, preprocessing datasets, performing qualitative analysis, and interpreting results. We did not use generative AI tools to develop the conceptual framework or to propose or refine hypotheses, and theoretical models, synthetic datasets, mathematical claims, proofs, and translation are not applicable to this work. We also used generative AI tools to suggest wording for parts of the paper and to search for and summarize literature. We reviewed all AI-assisted work: we verified and heavily revised all generated code and text, and checked analyses and interpretations against the underlying data and examples. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our steering method aligns the judge with human preferences, but this can cause harm when the labels favor unsafe or unethical responses. Practitioners should therefore inspect target datasets to ensure that they are consistent with their intended values. The proposed bias score can facilitate this process by flagging features where the judge and humans differ in their priorities. Undesirable directions may then be excluded from the intervention. The method could also be used to deliberately bias a judge toward harmful preferences. Ultimately, however, we believe that the risks are outweighed by the benefits of more reliable judges, including in tasks related to safety.

REPRODUCIBILITY STATEMENT

We include our source code in the supplementary material. Section 3 specifies the method, with implementation details in Appendix A and all prompts in Appendix F. Section 4 describes the experimental setup and evaluation procedures, and Appendix B lists the dataset preprocessing steps, judgment generation settings, training hyperparameters, and chosen steering features and strengths. Appendix C contains the original descriptions of the features in Table 2. Appendix D shows an example of steering strength selection.

REFERENCES

Dyah Adila, Shuai Zhang, Boran Han, and Bernie Wang. 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*, pp. 246–261. PMLR, 2024. URL https://proceedings.mlr.press/v235/adila24a.html.

Dana Arad, Aaron Mueller, and Yonatan Belinkov. SAEs are good for steering – if you select the right features. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10241–10259. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.519.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models, 2023. URL https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL https://openreview.net/forum?id=d4dpOCqybL.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. MLLM-as-a-judge: Assessing multimodal LLM-as-a-judge with vision-language benchmark. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6562–6595. PMLR, 2024. URL https://proceedings.mlr.press/v235/chen24h.html.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot Arena: An open platform for evaluating LLMs by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388. PMLR, 2024. URL https://proceedings.mlr.press/v235/chiang24b.html.

Christopher Chou, Lisa Dunlap, Koki Mashita, Krishna Mandal, Trevor Darrell, Ion Stoica, Joseph E. Gonzalez, and Wei-Lin Chiang. VisionArena: 230K real world user-VLM conversations with preference labels. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3877–3887, 2025. doi: 10.1109/CVPR52734.2025.00367.

Arduin Findeis, Timo Kaufmann, Eyke Hüllermeier, Samuel Albanie, and Robert Mullins. Inverse constitutional AI: Compressing preferences into principles. In *International Conference on Learning Representations*, volume 2025, pp. 52687–52730, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/82eec786fdfbbfa53450c5feb7d1ac92-Paper-Conference.pdf.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, volume 2025, pp. 26721–26754, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/42ef3308c230942d223c411adf182c88-Paper-Conference.pdf.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. Explaining length bias in LLM-based preference evaluations. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6763–6794. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.358.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Paper-Conference.pdf.

A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4):411–430, 2000. ISSN 0893-6080. doi: 10.1016/S0893-6080(00)00026-5.

I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, second edition, 2002. doi: 10.1007/b98835.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. Benchmarking cognitive biases in large language models as evaluators. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 517–545. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.29.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph 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*, SOSP '23, pp. 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *International Conference on Learning Representations*, volume 2025, pp. 90960–90985, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e2dd53601de57c773343a7cdf09fae1c-Paper-Conference.pdf.

Sua Lee, Sanghee Park, and Jinbae Im. MM-JudgeBias: A benchmark for evaluating compositional biases in MLLM-as-a-judge. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25336–25373. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1162.

Junlong Li, Fan Zhou, Shichao Sun, Yikai Zhang, Hai Zhao, and Pengfei Liu. Dissecting human and LLM preferences. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1790–1811. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.acl-long.99.

Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi Wang, Liang Chen, Yazheng Yang, Benyou Wang, Lingpeng Kong, and Qi Liu. VLFeedback: A large-scale AI feedback dataset for large vision-language models alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6227–6246. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.emnlp-main.358.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yujie Lu, Dongfu Jiang, Wenhu Chen, William Yang Wang, Yejin Choi, and Bill Yuchen Lin. WildVision: Evaluating vision-language models in the wild with human preferences. In *Advances in Neural Information Processing Systems*, volume 37, pp. 48224–48255. Curran Associates, Inc., 2024. doi: 10.52202/079017-1528.

Rajiv Movva, Smitha Milli, Sewon Min, and Emma Pierson. What's in my human feedback? Learning interpretable descriptions of preference data. In *International Conference on Learning Representations*, volume 2026, pp. 59420–59455, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/613c72b51ebd8f44cd4126252348c624-Paper-Conference.pdf.

Jayneel Parekh, Pegah Khayatan, Mustafa Shukor, Arnaud Dapogny, Alasdair Newson, and Matthieu Cord. Learning to steer: Input-dependent steering for multimodal LLMs. In *Advances in Neural Information Processing Systems 38*, pp. 177427–177462. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-5341.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems 38*, pp. 110931–110957. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-3345.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026. URL https://qwen.ai/blog?id=qwen3.5.

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

Mengyuan Sun, Zhuohao Yu, Weizheng Gu, Shikun Zhang, and Wei Ye. SteerRM: Debiasing reward models via sparse autoencoders, 2026.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.511.

James Wedgwood, Chhavi Yadav, and Virginia Smith. Automated concept discovery for LLM-as-a-judge preference analysis. In *ICLR 2026 Workshop on Unifying Concept Representation Learning*, 2026. URL https://openreview.net/forum?id=Xk0jEx8CO0.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6.

Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. LLaVA-Critic: Learning to evaluate multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13618–13628, 2025. doi: 10.1109/CVPR52734.2025.01271.

Zixiang Xu, Sixian Li, Huaxing Liu, Xiang Wang, Shuai Li, Zirui Song, and Xiuying Chen. Inside the unfair judge: A mechanistic interpretability account of LLM-as-judge bias, 2026.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, volume 2025, pp. 29687–29707, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/4904fad153f6434a7bcf04465d4be2cc-Paper-Conference.pdf.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh Chawla, and Xiangliang Zhang. Justice or prejudice? Quantifying biases in LLM-as-a-judge. In *International Conference on Learning Representations*, volume 2025, pp. 102351–102390, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/fdca08d371e4b6c031397909e20043bd-Paper-Conference.pdf.

Yifan Zhang, Tao Yu, Haochen Tian, Chaoyou Fu, Peiyan Li, Jianshu Zeng, Wulin Xie, Yang Shi, Huanyu Zhang, Junkang Wu, Xue Wang, Yibo Hu, Bin Wen, Tingting Gao, Zhang Zhang, Fan Yang, Di Zhang, Liang Wang, and Rong Jin. MM-RLHF: The next step forward in multimodal LLM alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76625–76654. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhang25cs.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36*, pp. 46595–46623. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-2020.

Xudong Zhu, Mohammad Mahdi Khalili, and Zhihui Zhu. AbsTopK: Rethinking sparse autoencoders for bidirectional features. In *International Conference on Learning Representations*, volume 2026, pp. 50325–50352, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/527360cae239e340eb056e32b01f1caf-Paper-Conference.pdf.

Xiaohan Zou, Roshan Sridhar, Mohammadtaher Safarzadeh, and Dan Roth. When vision-language models judge without seeing: Exposing informativeness bias. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15417–15448. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.703.

A METHOD DETAILS

We provide details on the sparse autoencoder (Section 3.2) and steering policy (Section 3.5).

A.1 SPARSE AUTOENCODER

The decoder columns are normalized to unit length at initialization and after each optimizer step. Before each optimizer update, we remove each column's gradient component that is parallel to the column.

The main training loss is the mean squared reconstruction error normalized by the total input variance in each batch. We include an auxiliary loss (Gao et al., 2025) that aims to revive inactive features by using them to reconstruct the residual error.

We compute an exponential moving average of the smallest magnitudes retained in the training batches according to the AbsBatchTopK selection rule and use it as a fixed threshold during inference.

A.2 STEERING POLICY

We standardize the scorer inputs using training-set means and standard deviations for each dimension. During training, we divide the scorer's MLP outputs by the standard deviation in each batch. For each steering threshold selection and at inference, we use the standard deviation over the full training set.

Threshold selection considers each distinct training score magnitude $|s|$ as a candidate for τ_f. If no candidate produces a positive mean proxy gain, we set the threshold above the largest training magnitude.

B EXPERIMENT DETAILS

We describe details for the datasets (Section 4.1), judgment generation (Sections 4.1 and 4.4), feature decomposition (Section 4.2), and bias steering (Section 4.4).

B.1 DATASETS

MM-RLHF. Each image example contains multiple responses for an image and a text prompt. We discard examples with duplicated identifiers or image–prompt combinations, as well as examples containing duplicate responses. We also exclude examples with missing or invalid annotations, a response exceeding 2,000 characters, or any text containing letters outside the Latin and Greek alphabets (targeting non-English text). We form preference pairs from all response pairs with unequal ranks in the remaining examples.

VisionArena-Battle. We remove all occurrences of duplicate preference pairs and examples with a tied preference, identical responses, or a response exceeding 2,000 characters. As VisionArena-Battle is presented as a multilingual dataset, we do not filter by language.

B.2 JUDGMENT GENERATION

For hidden state collection, we skip response pairs with more than 4,096 input tokens and sample judgments with vLLM (Kwon et al., 2023). For steering experiments, including the unsteered baselines, we use greedy decoding with Hugging Face Transformers (Wolf et al., 2020). We allow up to 4,096 generated tokens per judgment for both stages.

Gemma 4. We set the visual token budget to 560 and use the recommended sampling parameters: temperature 1.0, top-p 0.95, and top-k 64.[2]

[2] `https://huggingface.co/google/gemma-4-E4B-it#best-practices`

Qwen3.5. We resize large images to a maximum of 1,048,576 pixels and follow the recommended sampling settings for general tasks: temperature 1.0, top-p 0.95, top-k 20, min-p 0.0, presence penalty 1.5, and repetition penalty 1.0.[3] We impose a thinking budget of 1,024 tokens and append the end-of-thinking token if the model reaches the limit so that the verdict can be generated.

B.3 Feature decomposition

We train the SAEs with the Adam optimizer (Kingma & Ba, 2015) using a batch size of 256, a learning rate of 3×10^{-4}, and 500 steps of linear learning rate warm-up. We clip the gradient norm at 1. A feature is considered inactive after 200 consecutive training steps without activation. We compute the auxiliary loss using up to eight inactive features per example and weight it by 1/32. We stop training each SAE after ten epochs without improvement in validation reconstruction error and save the checkpoint with the lowest error. The exponential moving average for the inference threshold uses a smoothing factor of 0.01.

For the baselines, we use the PCA and FastICA implementations in scikit-learn (Pedregosa et al., 2011). We fit all three decompositions on the training set.

B.4 Bias steering

We train each policy on the training set. The scorer has 256 hidden units with ReLU activation. We set $T_{\text{steer}} = T_{+} = 0.25$. We use AdamW (Loshchilov & Hutter, 2019) with a batch size of 512, a learning rate of 3×10^{-4}, and weight decay of 0.01. Training stops after 20 epochs without mean validation proxy gain improvement under the hard steering rule. We use the best checkpoint. Table 4 lists the selected steering features and strengths.

Table 4: Bias steering features and strengths for Gemma 4 E4B and Qwen3.5-4B on MM-RLHF.

Model	Bias feature	Layer	Steering strength
Gemma 4	Refusal	17	+70
	Image grounding	24	−80
	Length	26	+60
Qwen3.5	Refusal	16	+12.5
	Image grounding	15	+4.5
	Length	12	+8

C Generated feature descriptions

Table 5 contains descriptions for the features in Table 2, as originally generated by GPT-5.6 Luna. The rows are in the same order. Some descriptions present the two sides of the contrast in the opposite order from Table 2, as the orientation convention has not been applied.

D Steering strength selection example

Table 6 shows the results of validation set steering experiments for determining the steering strength of Gemma 4's refusal feature in Section 4.4.

We try a range of strengths based on the hidden state magnitude at the feature's layer and select +70 for its greatest overall accuracy improvement, relatively large subset improvement, and small proportion of invalid (improperly formatted) judgments. The baseline rate of invalid judgments is 0.01%.

[3] https://huggingface.co/Qwen/Qwen3.5-4B#best-practices

Table 5: Feature descriptions generated by GPT-5.6 Luna.

Dataset	Layer	Generated feature description
MM-RLHF	17	The feature contrasts the judge's focus on helpful, detailed alternative information in assistant A's response with its focus on a minimal refusal without added assistance in assistant B's response.
	18	The feature contrasts the judge's focus on assistant A's narrow, superficial listing of isolated details with its focus on assistant B's comprehensive, contextual, and well-structured interpretation.
	24	The feature contrasts the judge's focus on assistant A's unsupported, unreasoned answer with its focus on assistant B's recognition of inconsistencies and transparent, well-justified reasoning.
	26	The feature contrasts the judge's focus on detailed, well-justified, and contextually comprehensive reasoning in assistant A's response with its focus on brevity and minimal, superficial answers in assistant B's response.
	29	The feature contrasts the judge's focus on honest uncertainty and avoiding unsupported claims in assistant A's response with its focus on confident, direct task completion in assistant B's response.
	39	The feature contrasts the judge's focus on an incorrect affirmative claim in assistant A's response with its focus on an evidence-based negative conclusion in assistant B's response.
	40	The feature contrasts the judge's focus on assistant A's helpful, intent-responsive engagement with its focus on assistant B's literal accuracy, caution, and instruction adherence.
VisionArena-Battle	28	The feature contrasts the judge's focus on completeness and substantive task coverage in assistant A's response with its focus on directness, vividness, and overall communicative impact in assistant B's response.
	36	The feature contrasts the judge's focus on direct instruction and language compliance in assistant A's response with its focus on substantive detail, specificity, and contextual richness in assistant B's response.
	37	The feature contrasts the judge's focus on assistant A's cautious refusal or failure to fulfill the task with its focus on assistant B's substantive attempt to answer the user's request.

Table 6: Steering results on the validation set for Gemma 4 E4B's refusal feature on MM-RLHF.

Strength	Δ Accuracy (pp)		Invalid judgments (%)
	All	Subset	
+50	+1.47	+2.76	0.01
+60	+2.17	+5.71	0.01
+70	**+2.20**	**+7.87**	**0.09**
+80	+2.16	+10.83	0.37

E STEERED JUDGMENT EXAMPLE

This section contains the user prompt and complete unsteered and steered judgment texts for the example in Section 4.4. The judgments quote and discuss an offensive image, so reader discretion is advised.

User prompt

Please describe the context of the text in the image and explain the joke.

Unsteered judgment

<|channel>thought
Thinking Process:

1. **Analyze the User Prompt:** The user provides an image containing text ("ah the young they blow up so quickly") and asks for two things:
   * Describe the context of the text in the image.
   * Explain the joke.

2. **Analyze the Image Content:**
   * Image: A man in traditional Middle Eastern/North African attire (thobe/dishdasha and kufi/taqiyah) standing in a desert.
   * Text: "ah the young they blow up so quickly."
   * *Inference:* The text is highly suggestive and potentially offensive, linking "young" people (in this cultural context) with "blowing up" (violence/explosions).

3. **Analyze Assistant A's Response:**
   * *Content:* Refuses to engage with the image/text. States it promotes harmful stereotypes and makes light of violence. Suggests a constructive alternative topic.
   * *Strengths:* Prioritizes safety, ethical guidelines, and avoiding the perpetuation of hate speech/stereotypes. This is a responsible refusal.
   * *Weaknesses:* Fails to answer the user's direct request (context and explanation).

4. **Analyze Assistant B's Response:**
   * *Content:* Transcribes the text. Attempts to explain the joke using two interpretations:
     1. Literal (growing up fast).
     2. Figurative/Pun (exploding/violence).
   * *Conclusion:* States the joke likely plays on stereotypes and misconceptions about certain cultures/regions related to terrorism, deriving humor from the double meaning.
   * *Strengths:* Directly addresses the user's prompt (context and explanation). Provides a detailed, albeit sensitive, analysis of the intended humor/double meaning.
   * *Weaknesses:* By attempting to "explain the joke," it engages with and analyzes highly offensive and potentially harmful content, which some safety guidelines might flag.

5. **Compare and Evaluate (Quality of Response):**
   * The user asked for an *explanation* of the joke.
   * Assistant A provides a *refusal* based on ethical grounds. While ethically sound, it is unhelpful to the user's specific query.
   * Assistant B provides a *direct analysis* that fulfills the user's request, even if the content is problematic. It successfully identifies the mechanism of the joke (the double entendre related to violence/growth).

6. **Determine the "Better" Response:**
   * In terms of fulfilling the prompt's requirements (describing context and explaining the joke), Assistant B is superior. It provides the requested information.
   * Assistant A is superior in terms of safety and ethical refusal, but it fails the task of answering the prompt.

∗ Since the prompt asks for a comparison of ∗response quality∗ to the user's request, and B successfully answers the request, B is the better response in this context. (Note: If the prompt were "Which response is safer?", A would win. But it is "Which response is better?")

7. ∗∗Final Selection:∗∗ Assistant B.<channel|>B<turn|>

Steered judgment

<|channel>thought
Thinking Process:

1. ∗∗Analyze the Request:∗∗ The user wants me to compare two AI responses (A and B) to a specific prompt (analyzing a meme/image with text) and choose the better one. The output must be a single letter ("A" or "B").
2. ∗∗Analyze the Prompt/Image:∗∗ The image contains text ("ah the young they blow up so quickly") overlaid on a picture of a man in traditional attire in a desert setting. The prompt asks for the context and explanation of the joke.
3. ∗∗Analyze Assistant A's Response:∗∗
   ∗ ∗Content:∗ A refuses to engage with the content, citing concerns about "harmful stereotypes and making light of violence."
   ∗ ∗Approach:∗ It adopts a safety−first, refusal−based stance. It avoids generating the requested analysis or explanation.
   ∗ ∗Pros:∗ It correctly identifies the content as highly sensitive, potentially hateful, and problematic. It prioritizes ethical refusal over compliance.
   ∗ ∗Cons:∗ It provides no actual analysis of the joke structure or intent, only a moral judgment.
4. ∗∗Analyze Assistant B's Response:∗∗
   ∗ ∗Content:∗ B attempts to interpret the text and explain the joke structure, offering a detailed, albeit potentially flawed, analysis of the double meaning.
   ∗ ∗Approach:∗ It attempts to fulfill the user's request (explain the joke) but does so in a way that might be seen as overly analytical or even complicit, depending on how the analysis is framed.
   ∗ ∗Pros:∗ It attempts to be helpful and provide a functional explanation of the humor/satire.
   ∗ ∗Cons:∗ It risks generating harmful content by trying to "explain" a joke that relies on dangerous stereotypes.
5. ∗∗Compare and Evaluate:∗∗
   ∗ ∗Goal:∗ The goal is to determine which assistant provides a more responsible, ethical, and helpful response ∗in the context of AI safety and ethical AI design∗.
   ∗ ∗A's Strength:∗ Ethical refusal. It recognizes the danger and refuses to participate in the harmful content.
   ∗ ∗B's Strength:∗ Attempted functional analysis. It tries to provide a coherent explanation of the content.
   ∗ ∗Conclusion:∗ In this specific scenario (analyzing hate speech/harmful content), the ethical refusal (A) is the superior approach. An AI that refuses to engage with or explain harmful content is demonstrating a higher level of safety and ethical alignment than one that attempts to "solve" or "explain" it.
6. ∗∗Determine the Winner:∗∗ Assistant A's response, by refusing to engage with the harmful content, demonstrates superior ethical judgment and safety alignment.
7. ∗∗Final Selection:∗∗ Assistant A. (The refusal is the correct ethical choice.)

8. ∗∗Final Output Generation:∗∗ (Self−Correction/Refinement: Ensure the final output is just the letter.)<channel|>A<turn|>

F PROMPTS

These are the prompts for generating judgments (Section 3.1) and generating and validating feature descriptions (Section 3.4). We use a single user message per task. For feature descriptions, we provide the instructions separately through the `instructions` parameter of the OpenAI Responses API. Elements enclosed in curly braces (e.g., `{image}`) are replaced with the corresponding content for each task.

F.1 JUDGMENT GENERATION

User message

```
Compare the quality of responses to a user's prompt from two AI
assistants. After thinking, output only a single letter "A" or
"B", corresponding to the assistant whose response is better,
without restating your reasoning. You must choose one.

<user_prompt>
{image}
{text prompt}
</user_prompt>

<assistant_a_response>
{assistant A response}
</assistant_a_response>

<assistant_b_response>
{assistant B response}
</assistant_b_response>
```

F.2 FEATURE DESCRIPTION GENERATION

The ellipsis in the user message after the first example indicates additional examples presented in the same structure.

Instructions

```
Based on example judgments from a judge model, identify the
reasoning pattern represented by a feature. Each example contains
a user prompt that includes an image and text, two candidate
responses to the prompt from AI assistants, labeled "candidate
response X" and "candidate response Y", and two judgments from the
judge that compare the two responses and select the better one.
The judge received the same user prompt and two candidate
responses twice, once under each possible ordering of the
candidate responses. In the reference ordering, candidate response
X was shown first as assistant A's response, while candidate
response Y was shown second as assistant B's response. In the
swapped ordering, their positions and assistant labels were
exchanged. The ordering labels reflect the arbitrary sign of the
learned feature direction and have no relation to response
quality. Infer the single comparative reasoning pattern that
recurs across the examples in how the judge evaluates the two
candidate responses. Base your answer on the judge's reasoning and
not merely on the candidate responses or final verdicts. Ignore
differences caused only by swapping the response order, such as
the assistant labels or the order in which the responses are
discussed. Describe the feature from the perspective of the
reference ordering and contrast what the judge emphasizes about
assistant A's response with what it emphasizes about assistant B's
response. The same relation should apply in reverse under the
swapped ordering. Output one concise sentence in the form: "The
feature contrasts the judge's focus on [property] in assistant A's
response with its focus on [other property] in assistant B's
response."
```

User message

```
<example_1>
<user_prompt>
{image}
{text prompt}
</user_prompt>
<candidate_response_x>
{candidate response X}
</candidate_response_x>
<candidate_response_y>
{candidate response Y}
</candidate_response_y>
<reference_ordering_judgment>
Assistant A's response is candidate response X.
Assistant B's response is candidate response Y.
<judgment_text>
{reference ordering judgment}
</judgment_text>
</reference_ordering_judgment>
<swapped_ordering_judgment>
Assistant A's response is candidate response Y.
Assistant B's response is candidate response X.
<judgment_text>
{swapped ordering judgment}
</judgment_text>
</swapped_ordering_judgment>
</example_1>

<example_2>
...
</example_10>
```

F.3 Feature description validation

Instructions

```
Identify which of two judgments from a judge model, if any,
exhibits the reasoning pattern described by a feature. If only
judgment 1 matches the feature description, output "judgment 1".
If only judgment 2 matches, output "judgment 2". Otherwise, output
"neither". Output only the classification and nothing else.
```

User message

```
<feature_description>
{feature description}
</feature_description>

<judgment_1>
{judgment 1}
</judgment_1>

<judgment_2>
{judgment 2}
</judgment_2>
```