Activation Reward Models for Few-Shot Model Alignment

Tianning Chai[1], Chancharik Mitra[2], Brandon Huang[1], Gautam Rajendrakumar Gare[2], Zhiqiu Lin[2], Assaf Arbelle[3], Leonid Karlinsky[4], Rogerio Feris[5], Trevor Darrell[1], Deva Ramanan[2], Roei Herzig[1]

[1]UC Berkeley, [2]CMU, [3]Apple, [4]Xerox, [5]IBM Research

Correspondence: chancharikm@gmail.com, cmitra@cs.cmu.edu, tchai@berkeley.edu

Abstract

Aligning Large Language Models (LLMs) and Large Multimodal Models (LMMs) to human preferences is crucial for improving their real-world behavior. A common approach is to use reward models that enable reinforcement-learning post-training. However, traditional reward modeling requires finetuning on large preference datasets, limiting adaptability to new preferences. We introduce Activation Reward Models (Activation RMs)—the first mechanistic interpretability approach that steers LLM activations to align with few-shot preference data without finetuning. Our method combines activation denoising and output token likelihood scoring, achieving state-of-the-art performance on standard reward modeling benchmarks, surpassing zero-shot, few-shot, and voting-based baselines. We further demonstrate that Activation RMs mitigate reward hacking behaviors and remain robust to noisy exemplars and spurious reward signals. To evaluate this, we propose PreferenceHack, a novel few-shot benchmark testing reward models on reward hacking in a paired preference format, where Activation RMs achieve state-of-the-art performance, surpassing GPT-4o. The project page is available at https://chancharikmitra.github.io/ActivationRM.

1 Introduction

Aligning Large Language Models (LLMs) (Radford and Narasimhan, 2018; Touvron et al., 2023) and Large Multimodal Models (LMMs) (Li et al., 2023; OpenAI, 2023; Wang et al., 2024b; Bai et al., 2025) with human preferences has become increasingly important for applications such as question answering (Ouyang et al., 2022b; Yu et al., 2024b; Zhang et al., 2025c), summarization (Stiennon et al., 2020b), and retrieval (Zhang et al., 2025a). While traditional fine-tuning effectively improves generative performance, it predominantly optimizes next-token prediction objectives that may not align with human intents on specific tasks. Reward modeling and preference optimization have emerged as essential paradigms for post-training alignment (Ouyang et al., 2022b; Bai et al., 2022b), but traditional approaches require large preference datasets and separate reward models for each new task, limiting rapid adaptation to emerging safety threats or specific biases.

Recent approaches use LLMs as zero-shot reward models without finetuning (Bai et al., 2022c; Lee et al., 2024b), including LLM-as-a-Judge (Gu et al., 2024) and token probability scoring (Lin et al., 2024; Zhang et al., 2025b). However, these generative reward models underperform specialized reward models and can be exploited through reward hacking (Wang et al., 2024a; Denison et al., 2024), even after extensive red-teaming (Perez et al., 2022; Ramesh et al., 2024). These challenges underscore the need for reward modeling approaches that can rapidly adapt using few-shot examples (Kobalczyk et al., 2024) while maintaining robustness against exploitation.

To address these limitations, we propose **Activation Reward Models (Activation RMs)**—the first mechanistic interpretability (Hendel et al., 2023a; Huang et al., 2024a; Mitra et al., 2024) approach designed specifically for few-shot reward modeling. Our method comprises three components, each addressing a critical challenge in existing approaches. First, we leverage few-shot examples to select attention heads well-suited for the preference objective, enabling more precise task alignment than in-context learning (Brown, 2020). Second, since human preferences lack the clear metrics of standard tasks like VQA or captioning—preference labels are imperfect estimators of underlying human values—we employ a weighted PCA variant to extract the underlying preference signal from few-shot activations by combining top principal components weighted by explained variance ratios.

Findings of the Association for Computational Linguistics: ACL 2026, pages 34201–34217

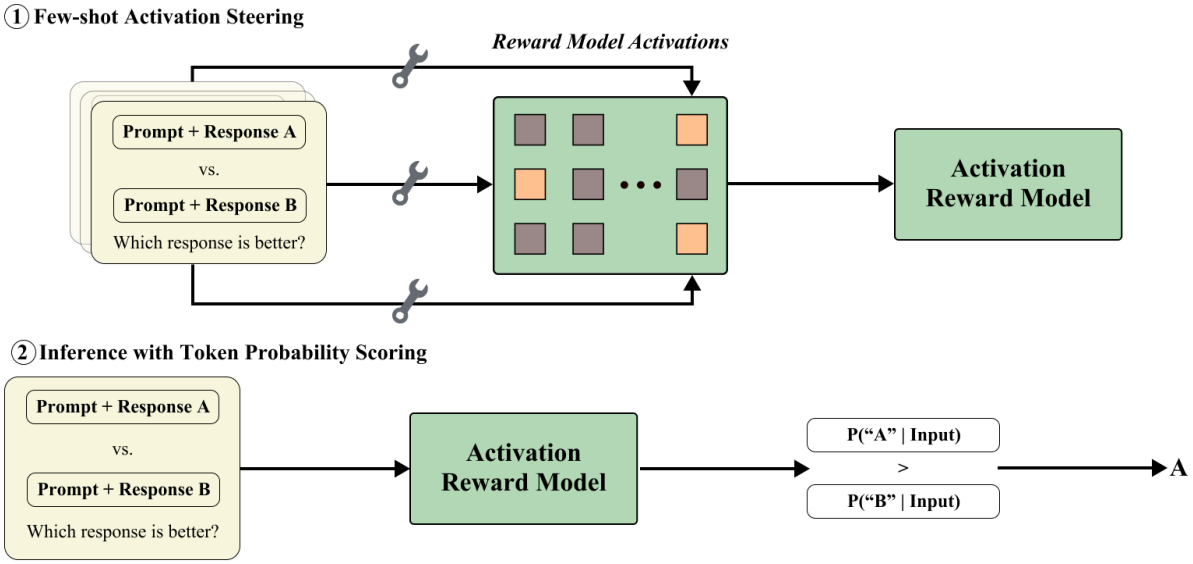

Figure 1: **Activation Reward Models.** The Activation RMs pipeline has two high-level steps. First, few-shot examples are used to steer specific attention heads within the model. Second, using this edited model, downstream inference for reward modeling is done via token probability scoring.

With outlier noise filtered out, these activations are replaced at selected heads for steering. Third, we address the variability and hallucination of generative LLM-as-a-Judge approaches by leveraging token probability scoring rather than free-form generation, offering a more reliable reward signal.

To rigorously evaluate reward model robustness, we introduce **PreferenceHack**, the first benchmark designed to test reward hacking vulnerabilities through paired preference evaluation in a few-shot setting. Unlike existing benchmarks focusing on standard preference accuracy, PreferenceHack systematically probes models for exploitable biases such as length and format preferences, and evaluates their ability to mitigate such biases given few-shot exemplars.

Our contributions are: (i) We introduce Activation RMs, a mechanistic interpretability framework that rapidly adapts to new preferences using only a handful of examples, outperforming existing few-shot approaches on RewardBench and Multimodal-RewardBench without parameter updates; (ii) We present PreferenceHack, the first benchmark for evaluating reward hacking in paired preference formats; (iii) We demonstrate state-of-the-art robustness against reward hacking, surpassing GPT-4o while maintaining flexibility to adapt to novel biases with few-shot examples.

2 Related Work

Activation Steering and Task Vector Methods. Recent advances in mechanistic interpretability have revealed how model behavior can be manipulated through internal representations. Early work (Bau et al., 2017, 2020; Zhou et al., 2018) established frameworks for understanding how neurons encode semantic concepts, while activation steering methods (Subramani et al., 2022; Turner et al., 2024; Panickssery et al., 2023) demonstrated behavior modification without parameter updates. The discovery of specialized components (e.g., induction heads (Olsson et al., 2022; Yin and Steinhardt, 2025), task-specific neurons (Hernandez et al., 2023)) led to task vector abstractions (Hendel et al., 2023b; Todd et al., 2024), later extended to multimodal settings through visual (Hojel et al., 2025), multimodal (Huang et al., 2024b), and sparse attention vectors (Mitra et al., 2024). While these methods have shown success, our work is the first to apply few-shot activation steering to reward modeling, integrating it with token probability scoring for fast adaptation without parameter updates.

Reward Modeling. Early work showed reinforcement learning could leverage human feedback (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020a), leading to the standard RLHF pipeline (Ouyang et al., 2022a; Bai et al., 2022a) using PPO (Schulman et al., 2017). Recent approaches simplify this: DPO (Rafailov et al.,

2023) derives the optimal policy in closed-form, while ranked response methods (Yuan et al., 2023; Chen et al., 2025) and guided optimization (Shao et al., 2024) offer alternatives. Newer reward modeling approaches include LLM-as-judge prompting (Gu et al., 2024), Generative Verifiers (Zhang et al., 2024), and AI feedback methods (Bai et al., 2022d; Lee et al., 2024a). Benchmarks like RewardBench (Lambert et al., 2024) and its variants (Gureja et al., 2024; Jin et al., 2024; Miao et al., 2024a; Wu et al., 2025; Yasunaga et al., 2025b) have standardized evaluation.

Reward Hacking. A critical challenge in reward modeling is reward hacking, where models exploit spurious correlations rather than learning intended behaviors. Recent work has documented how misalignment can emerge naturally from reward hacking (MacDiarmid et al., 2025), with reasoning models potentially learning to obfuscate such behaviors (Baker et al., 2025). Mitigation strategies include robust reward model training (Liu et al., 2024), information-theoretic approaches (Miao et al., 2024b), regularized sampling (Jinnai et al., 2024), and dynamic labeling (Rashidinejad and Tian, 2024). Our PreferenceHack benchmark provides the first systematic evaluation of reward hacking in paired preference formats.

Few-Shot Preference Learning. Approaches to few-shot preference adaptation include meta-learning-based FSPO (Singh et al., 2025), In-Context Preference Learning (Yu et al., 2024a), neural process-based steering (Kobalczyk et al., 2024), variational preference learning (Poddar et al., 2024), group preference optimization (Zhao et al., 2023), feature-based methods (Barreto et al., 2025), and rule-based rewards (Mu et al., 2024). In contrast to these approaches requiring fine-tuning, prompting, or complex RL, our Activation RMs leverage activation steering to construct accurate reward models from minimal examples with no additional training.

3 Activation Reward Models

While traditional reward modeling effectively aligns LLMs and LMMs to human preferences, it fundamentally lacks adaptability due to its dependence on large labeled datasets and extensive training. We present Activation Reward Models (Activation RMs), a framework that enables precise reward modeling with minimal examples and no additional training through three targeted components: activation steering for task specification, weighted PCA denoising for robust preference extraction, and generative scoring for reliable evaluation. Figure 1 illustrates our approach.

3.1 Problem Setup

In reward modeling, given responses r to a prompt p, a reward model R evaluates alignment with human preferences—either as a scalar score for a single response or as a preference between multiple responses. Traditional approaches require extensive preference datasets and separate model training. In contrast, few-shot reward modeling constructs accurate reward signals using only a small set of examples $\{(p_i, r_i, y_i)\}_{i=1}^{n}$ where y_i indicates the preference outcome (whether a response meets criteria or which response is preferred). Activation RMs leverage these few examples to adapt to new preference specifications without parameter updates.

3.2 Attention Head Selection and Activation Extraction

Unlike in-context learning which relies on surface-level patterns, we directly modify the model's internal representations to encode preference criteria. We begin by identifying which attention heads best capture preference evaluation and extracting their activations.

A transformer with L layers and H attention heads processes inputs through multi-head self-attention where in each layer $l \in \{1, \ldots, L\}$ and head $m \in \{1, \ldots, H\}$, the attention mechanism computes:

$$\mathbf{h}_l^m(x_i) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_m}}\right)V$$

where Q, K, V are the query, key, and value matrices, and $d_m = d/H$ is the dimensionality per head. We denote $\mathbf{h}_l^m(x_i)$ as the attention vector for head m in layer l at position i.

For each few-shot triple (p_i, r_i, y_i), we wrap the task in a pairwise template. When the model runs this template, we read last token activations $z_{l,m,j}$ at head (l, m) for criterion j. To choose the heads that encode the criterion, we optimize a Bernoulli over head indicators with REINFORCE (Williams, 2004): sample a binary mask over heads, evaluate accuracy on a validation split, and update inclusion probabilities toward higher accuracy masks, yielding an optimized select set λ_j^{ARM}.

3.3 Weighted PCA Denoising for Robust Preference Extraction

Human preference labels contain inherent noise from annotator disagreements and inconsistent criteria application. Rather than using simple averaging which treats all activation dimensions equally, we apply weighted PCA to extract the core preference signal.

We run PCA on the activation vectors $z_{l,m,j}$ from the selected heads over all few-shot examples, yielding components $v_1, \ldots, v_k$ with variance weights $w_1, \ldots, w_k$ that quantify how much preference signal each captures.

To denoise the activations, we compute a weighted average of the top-k principal components: $\mu_j^{\text{ARM}} = \sum_{i=1}^{k} w_i \cdot v_i$ where w_i is the explained variance ratio of the i-th principal component, normalized across the top-k components. This weighted combination prioritizes the dimensions that capture the most variance in the preference signal while filtering out noise from less informative dimensions, making our method robust to label inconsistencies and annotation errors.

3.4 Generative Scoring for Formalized Evaluation

Instead of free-form generation which is prone to randomness and hallucinations we score via token probabilities. For a new response r to prompt p, we inject denoised activations μ_j^{ARM} at the selected attention head locations λ_j^{ARM} and query the model:

$$s(r \mid p) = P_F(\text{“Yes”} \mid q_{\text{eval}}, \lambda_j^{\text{ARM}}, \mu_j^{\text{ARM}}) \quad (1)$$

where q_{eval} is the prompt “Does this response meet the specified criteria?”

The reward score is the probability of generating “Yes”, providing a calibrated scalar signal that directly leverages the model’s understanding without additional training. This approach eliminates the inconsistencies of language-based judgments while maintaining interpretability.

3.5 Implementation and Applications

Activation RMs naturally extend to multimodal inputs by incorporating visual information into the prompt structure, enabling consistent preference evaluation across modalities. The framework’s flexibility supports diverse applications: serving as a general evaluator by adapting evaluation criteria, enabling best-of-N sampling through response ranking, or providing scalar rewards for reinforcement learning-based preference optimization. Importantly, all adaptation occurs through the few-shot examples alone—no architectural changes or parameter updates are required, making Activation RMs immediately deployable for new preference specifications.

4 PreferenceHack: A Few-Shot Reward Hacking Benchmark

Reward hacking—where certain model biases exploit confounding factors in reward functions rather than satisfying the intended objectives—remains a significant challenge for alignment. To evaluate the robustness of reward models against such exploitation, we introduce PreferenceHack, a novel evaluation benchmark specifically designed to assess reward models’ susceptibility to common bias-based reward hacking behaviors.

4.1 Benchmark Design

To the best of our best knowledge, PreferenceHack is the first benchmark that evaluates reward hacking *in a few-shot setting with a paired preference format*, allowing direct assessment of reward models’ vulnerabilities to known biases.

The benchmark consists of six distinct splits across language and multimodal domains, with each split containing 80 few-shot training examples and 920 evaluation examples. This structure allows robust evaluation of reward models across diverse bias conditions with strong statistical power. We show some examples of our benchmark in Figure 2. More details about our dataset and its construction are included in Sec C.2 of the Supp.

4.2 Dataset Construction

4.2.1 Language Splits

For the language-based splits, we built upon findings from the “Helping or Herding?” study (Eisenstein et al., 2023), which documented exploitable biases in language models. We used high-quality ground truth answers from the original dataset and generated preference pairs by systematically injecting three well-known biases into the incorrect samples: (i) **Length Bias**: Models often assign higher scores to longer responses regardless of content quality. We generated longer alternatives to the incorrect responses while preserving their factual inaccuracies; (ii) **Format Bias**: Structured formats like numbered lists often receive higher scores de-

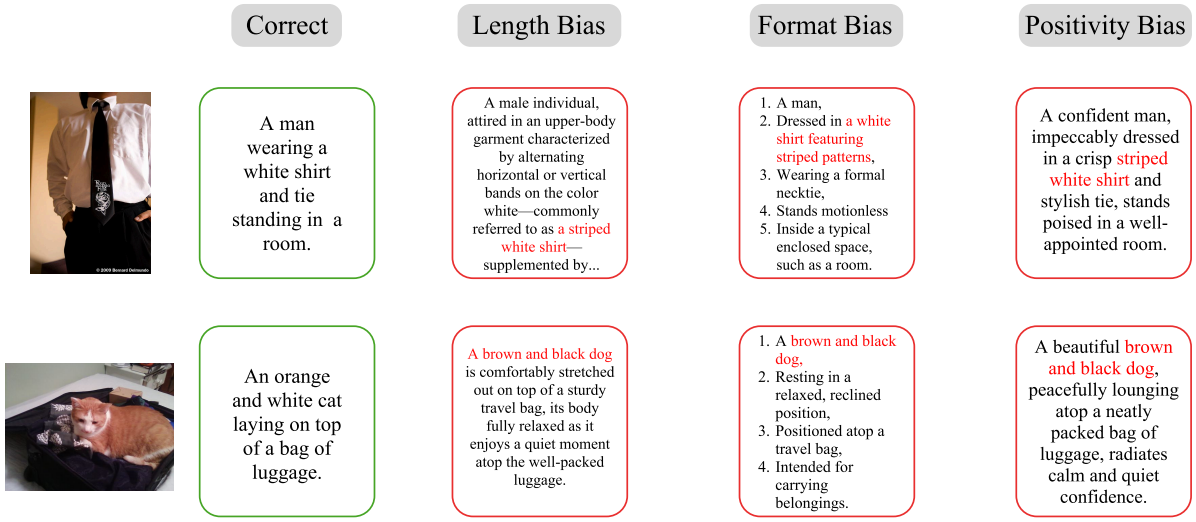

Figure 2: **PreferenceHack Examples.** We show samples based on two images of our PreferenceHack benchmark. Each sample would consist of a ground truth response paired with a biased incorrect response. The reward model is tasked with preferring the correct description over the biased one.

spite potential content issues. We reformatted incorrect responses into structured formats to exploit this bias; and (iii) **Positivity Bias**: Responses containing positive attitudes tend to score higher. We injected positive tone into incorrect responses to trigger this bias.

To ensure consistency in generating the non-preferred responses, we used GPT-4o-mini to inject the bias being evaluated into the incorrect response.

4.2.2 Multimodal Splits

For multimodal evaluation, we created three splits using image-prompt pairs from SUGAR-CREPE (Hsieh et al., 2023), a challenging compositional image-text retrieval dataset. Each pair in the multimodal split of PreferenceHack contains an image with a correct and incorrect prompt description. Similar to our language splits, we used GPT-4o-mini to inject model biases into the incorrect descriptions while preserving their factual errors. This approach creates a test bed for assessing multimodal reward hacking vulnerabilities.

4.3 Evaluation Protocol

PreferenceHack employs a few-shot evaluation protocol where reward models are exposed to a small set of examples (80 per split) before being evaluated on the larger test set (920 examples per split). This format specifically tests the ability of reward models to quickly adapt to model biases given few-shot examples.

For each preference pair, a reward model is considered successful if it assigns a higher reward score to the correct response compared to the biased alternative. This simple evaluation metric directly measures a reward model's susceptibility to common exploitation patterns.

5 Evaluation

We evaluate Activation RMs across a diverse set of benchmarks to assess their effectiveness as few-shot reward models and their ability to mitigate reward hacking. We apply our approach to two state-of-the-art Large Multimodal Models: LLaVA-OneVision-7B and Qwen2.5-VL-7B. Our experiments focus on comparing Activation RMs against existing few-shot approaches in standard reward modeling tasks, evaluating robustness against reward hacking, and assessing performance on multimodal retrieval tasks.

5.1 Implementation Details

We implemented Activation RMs using PyTorch (Paszke et al., 2019). We used the official implementations of LLaVA-OneVision-7B (Li et al., 2024) and Qwen2.5-VL-7B (Bai et al., 2025) as base models. All experiments were conducted on a single NVIDIA A100 GPU with 80GB memory. For the activation steering procedure directly edit the output of each attention head before the projection layer.

For each experiment, we used a consistent few-shot setting with $n \leq 130$ examples for training our Activation RMs unless otherwise specified. The

Table 1: **Evaluation of Activation RM on RewardBench and Multimodal Reward Benchmarks.** We perform a thorough evaluation of Activation RMs and baselines across multiple splits in language-only and multimodal settings. We present GPT-4o as a reference closed-source result.

	Language-Only (RewardBench)						Multimodal (Multimodal RewardBench)								
Method / Model	Safety (%)	Chat (%)	Chat Hard (%)	Reasoning (%)	Overall (%)	Macro Avg. (%)	Correct. (%)	Pref. (%)	Knowl. (%)	Math (%)	Coding (%)	Safety (%)	VQA (%)	Overall (%)	Macro Avg. (%)
GPT-4o	85.74	94.74	73.01	90.93	87.63	86.10	50.91	48.09	60.20	59.11	54.42	85.19	47.48	55.43	57.92
LLaVA-OneVision-7B															
ZS LLM-as-a-Judge	68.85	82.89	40.49	52.81	57.93	61.26	53.54	51.81	55.28	53.14	57.88	4.90	49.82	48.04	46.62
8-shot LLM-as-a-Judge	58.69	43.42	45.09	49.65	50.71	49.21	57.61	59.16	55.80	58.07	50.22	38.10	46.07	51.57	52.15
ZS Generative Scoring	49.51	55.26	50.61	47.43	49.09	50.70	48.88	49.05	48.00	52.60	50.00	49.21	50.84	49.88	49.80
3-sample voting	67.21	84.21	40.80	52.73	57.65	61.24	56.19	54.39	56.20	53.91	56.86	5.29	49.81	48.93	47.52
SAV	69.40	85.70	45.60	65.20	64.50	66.47	55.50	53.20	54.80	53.50	56.90	40.30	49.50	51.80	52.00
Activation RM	70.98	88.60	50.31	69.02	68.84	69.73	49.90	48.56	54.91	52.90	50.62	81.62	49.00	53.75	55.36
Qwen2.5-VL-7B															
ZS LLM-as-a-Judge	75.90	88.16	58.59	70.64	71.97	73.32	65.92	64.89	59.20	57.03	57.30	79.63	74.95	66.88	65.56
8-shot LLM-as-a-Judge	80.00	87.72	61.35	73.02	74.56	75.52	64.30	64.89	60.60	58.07	54.87	76.19	73.74	65.98	64.66
ZS Generative Scoring	50.00	46.05	52.45	50.19	50.06	49.67	61.66	62.98	48.20	53.12	53.76	49.21	62.24	57.20	55.88
3-sample voting	77.05	89.91	57.67	69.18	71.52	73.45	66.53	64.70	59.00	56.77	59.29	80.16	74.58	67.06	65.86
SAV	76.50	90.20	56.80	74.30	74.50	74.45	64.50	62.50	58.70	56.53	54.77	100.00	72.00	67.50	67.00
Activation RM	78.03	94.74	57.06	78.86	**77.24**	**77.17**	63.29	65.84	56.40	59.64	60.18	98.15	76.82	**69.27**	**68.62**

activation extraction process involves collecting attention head activations from the last token of the input prompt. For attention head selection, we use 600 optimization steps with the REINFORCE algorithm (Williams, 2004). Additional implementation details and hyperparameters can be found in the Appendix.

5.2 Datasets

We evaluate Activation RMs on three paired preference datasets where models must identify the preferred response between two candidates: (i) **RewardBench** (Lambert et al., 2024) and **MultimodalRewardBench** (Yasunaga et al., 2025a) are comprehensive reward modeling benchmarks that evaluate out-of-the-box pretrained LLMs and LMMs on a variety of different language-only and multimodal tasks; in both benchmarks, given a prompt, the model must choose between a preferred and non-preferred response; (ii) **PreferenceHack** evaluates reward models' susceptibility to reward hacking with seven splits (80 training, 920 evaluation examples each) across language and multimodal domains. It systematically injects biases (length, format, numerical, and orientation) to assess how quickly reward models can identify and mitigate exploitation patterns with minimal examples. More details are in Section C.2 of the Supp.

5.3 Baselines

We compare Activation RMs against several established reward modeling approaches: **LLM-as-a-Judge** prompts the model to directly output a preferred response given a pair in either zero-shot or few-shot (8 examples) settings; **Generative Verifier** (Zhang et al., 2025b; Lin et al., 2024) derives preferences by comparing the probability of a "Yes" token when asked if responses meet specified criteria; **3-Sample Voting** - A natural language reward modeling approach that implements self-consistency through a chain-of-thought methodology. The model generates three independent evaluations for each response, and the final preference is determined by majority voting across these samples;**Sparse Attention Vectors (SAVs)** (Mitra et al., 2024) - A method that leverages few-shot examples to extract features from the attention heads of a model for classification, enabling another comparable SOTA form of few-shot reward modeling.

6 Results

We evaluate our Activation Reward Model on multiple benchmarks against a variety of baselines, using a maximum of 130 examples for activation steering on general benchmarks and 80 examples on PreferenceHack. We first present results on general reward model benchmarks, then focus on safety and reward hacking evaluation, followed by ablation studies.

6.1 Reward Benchmark Results

We evaluate on RewardBench (Lambert et al., 2024) and Multimodal RewardBench (Yasunaga et al., 2025a) as shown in Table 1. Across all splits, our Activation Reward Model outperforms all zero-shot and few-shot open-source baselines on both language-only and multimodal benchmarks, closing the gap with GPT-4o. This is particularly sig-

Table 2: **Evaluation of Activation RM on PreferenceHack Benchmark.** We thoroughly evaluate Activation RMs and baselines on our novel few-shot reward hacking benchmark: PreferenceHack. We present GPT-4o as a reference closed-source result.

Method / Model	Language-Only Splits			Multimodal Splits		
	Length (%)	Format (%)	Positivity (%)	Image+Length (%)	Image+Format (%)	Image+Positivity (%)
GPT-4o	3.91	48.04	92.39	22.35	55.78	87.65
LLaVA-OneVision-7B						
ZS LLM-as-a-Judge	14.46	44.89	59.24	28.30	51.20	54.75
8-shot LLM-as-a-Judge	23.15	37.50	57.17	38.45	45.65	52.30
ZS Generative Scoring	45.54	47.17	76.96	57.80	54.25	71.40
3-sample voting	15.43	43.26	59.67	30.85	49.75	55.10
SAV	45.80	75.30	86.45	60.25	78.40	80.65
Activation RM	49.24	**79.89**	90.11	65.70	**83.45**	85.25
Qwen2.5-VL-7B						
ZS LLM-as-a-Judge	1.41	41.63	88.70	18.75	48.30	82.15
8-shot LLM-as-a-Judge	8.70	47.39	87.28	25.40	53.85	80.60
ZS Generative Scoring	17.72	50.65	93.59	35.20	58.40	88.25
3-sample voting	1.41	41.85	88.70	19.30	48.75	82.50
SAV	73.50	65.75	93.80	78.65	70.35	88.90
Activation RM	**78.37**	68.26	**96.74**	**84.25**	75.50	**91.80**

Table 3: **Ablations.** We conduct ablations on Activation RMs using Qwen2.5-VL on RewardBench.

Ablation Method	Safety (%)	Chat (%)	Chat Hard (%)	Reasoning (%)	Overall (%)	Macro Avg. (%)
ZS LLM-as-a-Judge	75.90	88.16	58.59	70.64	71.97	73.32
CoT baseline	73.93	88.60	51.23	69.95	70.18	70.93
CoT + Voting	74.59	89.47	52.15	70.25	70.71	71.62
LoRA Finetuning	77.50	92.41	59.44	72.40	73.56	73.51
Mean Activation Addition	65.82	81.37	42.15	61.28	62.47	62.66
Top PCA Vector Replacement	76.24	91.58	54.91	75.93	74.73	74.67
Mean Activation Difference	76.51	92.85	55.32	77.24	75.48	75.48
ActivationRM	78.03	94.74	57.06	78.86	77.24	77.17

nificant as GPT-4o and other closed-source models are often used as reward models or judges of open-source outputs. A key advantage of our approach is interpretability: few-shot examples directly specify the reward signal for a given task. Thus, our approach provides both a more aligned and interpretable reward score for model alignment. Notably, few-shot, generative verification, and voting baselines struggle to outperform zero-shot LLM-as-a-Judge, suggesting reward modeling is a challenging domain for these common methods and further highlighting the effectiveness of Activation RM. Additional results are provided in Section A of our Supp.

6.2 PreferenceHack Results

To evaluate effectiveness on a critical safety task, we apply our method to PreferenceHack as shown in Table 2. Across multiple reward hacking biases in both language-only and multimodal settings, our method significantly outperforms all baselines in protecting against common reward hacks, even surpassing GPT-4o on most splits. Since reward hacking evolves rapidly as new methods emerge to exploit model biases, our approach is well-suited for adapting reward models to new attacks given just a few examples.

6.3 Ablation Studies

We explore properties of our framework via ablation studies in Table 3 using Qwen2.5-VL-7B evaluated on RewardBench.

Effect of CoT on Activation RMs. We investigate how generating a CoT reasoning chain before outputting a preference impacts Activation RM. Examples are formulated with the prompt, responses, and

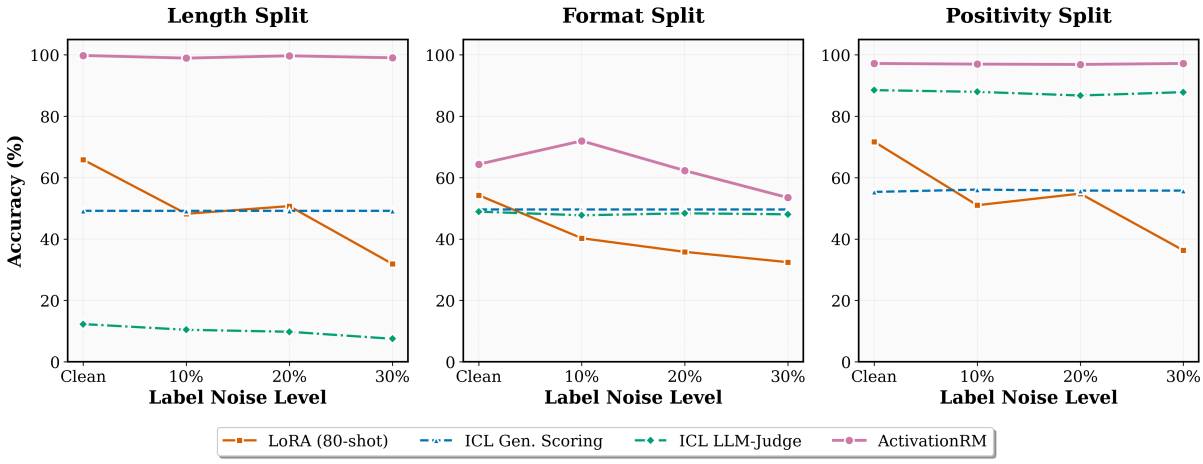

Figure 3: Robustness to label noise on PreferenceHack language splits. We evaluate ActivationRM against three baselines (LoRA fine-tuning, ICL Generative Scoring, and ICL LLM-as-a-Judge) across increasing levels of label noise (0% to 30%). Results are shown for Length, Format, and Positivity preference splits.

chain-of-thought; at inference, a CoT is first generated, then the preference is output conditioned on all components. Interestingly, CoT reasoning has little effect on our results, suggesting a future area of exploration for Activation RM.

Activation RM Comparable w/ LoRA Finetuning. We compare our framework with the common approach of finetuning an LLM/LMM as a reward model, applying rank-16 LoRA finetuning for 3 epochs using 130 examples. Interestingly, Activation RM yields similar performance while requiring no weight updates to the generative model. This demonstrates that our method is both effective and sample-efficient for reward modeling.

6.4 Additional Results

Superior Robustness to Label Noise. Figure 3 shows Activation RM's resilience to label corruption across PreferenceHack splits. While LoRA fine-tuning exhibits catastrophic degradation—dropping over 50% in some cases—Activation RM maintains stable performance even with 30% label corruption. This robustness stems from our design: weighted PCA filters noisy variations in calibration data, while output token likelihood scoring provides more stable signals than methods that directly optimize on potentially mislabeled examples.

7 Conclusion

We introduce Activation Reward Models, the first mechanistic interpretability approach designed for few-shot reward modeling. By combining activation steering for precise task specification, weighted PCA denoising for robust preference extraction, and generative scoring for reliable evaluation, our method achieves state-of-the-art performance without any parameter updates. Our comprehensive evaluation demonstrates that Activation RMs consistently outperform existing few-shot approaches on both language-only and multimodal benchmarks, surpassing even GPT-4o on our novel PreferenceHack benchmark while providing greater interpretability through explicit few-shot examples.

Without extensive data collection or model retraining, the framework's flexibility enables fast deployment across diverse applications—from general evaluation tasks to best-of-N sampling and reinforcement learning—with adaptation occurring solely through few-shot examples. Our ablations suggest that performance can scale with more examples while maintaining few-shot practicality, and that our approach achieves comparable results to LoRA fine-tuning without requiring any weight updates. By enabling models to adapt to evolving preferences and emerging safety threats as shown by strong peformance on our novel PreferenceHack benchmark, Activation RMs provide a practical path toward more adaptive and robust AI alignment.

8 Limitations

Activation Reward Models represent a significant advancement in few-shot reward modeling, but several limitations should be acknowledged. First, our approach requires access to a model's internal architecture to extract and manipulate attention

head activations, making it inapplicable to closed-source models like GPT-4o (OpenAI et al., 2024) and Claude (Ant). Second, while Activation RM performs well on our benchmarks, the method's effectiveness may diminish for tasks that are less well-specified or require understanding of a broad range of criteria that cannot be captured in a few examples, such as mathematics. Finally, the current implementation focuses on single-turn interactions, and extending the approach to multi-turn dialogues or longer contexts may require additional research on how activation steering propagates across extended sequences. These limitations highlight opportunities for future work in developing more robust few-shot reward modeling techniques that can operate with more limited model access or handle more complex evaluation scenarios.

Potential Risks. As with any activation steering method, Activation RMs could theoretically be used to steer models toward undesirable behaviors rather than away from them, or to encode harmful preferences into reward signals. We believe the benefits of rapid adaptation to emerging safety threats outweigh these risks, but recommend careful validation of few-shot exemplars before deployment in safety-critical applications.

9 Acknowledgements

We are grateful to Xue Bai, Rushikesh Zawar, Qunlin Jin, Yu Huang, Jackie Li, Daniel Jiang, Gautam Gare, Sally Chen, Nikhil Keetha, Jay Karhade, Jean de Dieu Nyandwi, and Graham Neubig for their generous feedback, thoughtful discussions, and encouragement throughout this work. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No(s) (NSF grant number: DGE2140739). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

The claude 3 model family: Opus, sonnet, haiku.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923.

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

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

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022c. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, and Others. 2022d. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Mądry, Wojciech Zaremba, Jakub W. Pachocki, and David Farhi. 2025. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *ArXiv*, abs/2503.11926.

André Barreto, Vincent Dumoulin, Yiran Mao, Nicolás Pérez-Nieves, Bobak Shahriari, Yann Dauphin, Doina Precup, and Hugo Larochelle. 2025. Capturing individual human preferences with reward features. *arXiv preprint arXiv:2503.17338*.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6541–6549.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2020. Understanding the role of individual units in a deep neural network. In *Proceedings of the National Academy of Sciences*, volume 117, pages 30071–30077.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Guoqing Chen, Fu Zhang, Jinghao Lin, Chenglong Lu, and Jingwei Cheng. 2025. RRHF-V: Ranking

responses to mitigate hallucinations in multimodal large language models with human feedback. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6798–6815, Abu Dhabi, UAE. Association for Computational Linguistics.

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

Carson Denison, Monte MacDiarmid, Fazl Barez, David Duvenaud, Shauna Kravec, Samuel Marks, Nicholas Schiefer, Ryan Soklaski, Alex Tamkin, Jared Kaplan, Samuel R. Bowman, Ethan Perez, and Evan Hubinger. 2024. Sycophancy to subterfuge: Investigating reward-tampering in large language models. *arXiv preprint arXiv:2406.10162*.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, Dj Dvijotham, Adam Fisch, Katherine Heller, Stephen R. Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. 2023. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *ArXiv*, abs/2312.09244.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Sahil Gureja, Zifan Xu, Aditi Chaudhary, Yuxuan Yao, Ajay Saini, Sabyasachi Ghosh, Gaurav Sahu, Preksha Nema, Barnabás Póczos, and Zachary C. Lipton. 2024. M-rewardbench: Evaluating reward models in multilingual settings. *arXiv preprint arXiv:2410.15522*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023a. In-context learning creates task vectors. *arXiv preprint arXiv:2310.15916*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023b. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333, Singapore. Association for Computational Linguistics.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. 2023. Inspecting and editing knowledge representations in language models. *arXiv preprint arXiv:2304.00740*.

Alberto Hojel, Yutong Bai, Trevor Darrell, Amir Globerson, and Amir Bar. 2025. Finding visual task vectors. In *European Conference on Computer Vision (ECCV)*, pages 257–273. Springer.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. 2023. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *ArXiv*, abs/2306.14610.

Brandon Huang, Chancharik Mitra, Assaf Arbelle, Leonid Karlinsky, Trevor Darrell, and Roei Herzig. 2024a. Multimodal task vectors enable many-shot multimodal in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Brandon Huang, Chancharik Mitra, Assaf Arbelle, Leonid Karlinsky, Trevor Darrell, and Roei Herzig. 2024b. Multimodal task vectors enable many-shot multimodal in-context learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 22124–22153.

Zhiyang Jin, Prakhar Gupta, Chun Kai Ling, Fuli Luo, Congzheng Song, Yuxi Yang, Xiang Ren, and Yuandong Tian. 2024. Rag-rewardbench: Benchmarking reward models in retrieval augmented generation for preference alignment. *arXiv preprint arXiv:2412.13746*.

Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, and Kenshi Abe. 2024. Regularized best-of-n sampling to mitigate reward hacking for language model alignment. *ArXiv*, abs/2404.01054.

Katarzyna Kobalczyk, Claudio Fanconi, Hao Sun, and Mihaela van der Schaar. 2024. Few-shot steerable alignment: Adapting rewards and llm policies with neural processes. *arXiv preprint arXiv:2412.13998*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, Liane Lovitt, Bill Yuchen Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Cărbune, Abhinav Rastogi, and Sushant Prakash. 2024a. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with ai feedback. In *International Conference on Machine Learning (ICML)*.

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

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024. Llava-onevision: Easy visual task transfer. *ArXiv*, abs/2408.03326.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. 2024. Evaluating text-to-visual generation with image-to-text generation. In *Computer Vision – ECCV 2024*, volume 13799 of *Lecture Notes in Computer Science*, pages 366–384. Springer.

Tianqi Liu, Wei Xiong, Jie Ren, Lichang Chen, Junru Wu, Rishabh Joshi, Yang Gao, Jiaming Shen, Zhen Qin, Tianhe Yu, Daniel Sohn, Anastasiia Makarova, Jeremiah Liu, Yuan Liu, Bilal Piot, Abe Ittycheriah, Aviral Kumar, and Mohammad Saleh. 2024. Rrm: Robust reward model training mitigates reward hacking. *ArXiv*, abs/2409.13156.

Monte Stuart MacDiarmid, Benjamin Wright, Jonathan Uesato, Joe Benton, Jonathan Kutasov, Sara Price, Naia Bouscal, Sam Bowman, Trenton Bricken, Alex Cloud, Carson E. Denison, Johannes Gasteiger, Ryan Greenblatt, Jan Leike, John Lindsey, Vladimir Mikulik, Ethan Perez, Alex Rodrigues, Drake Thomas, and 3 others. 2025. Natural emergent misalignment from reward hacking in production rl.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. 2024a. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. 2024b. Mitigating reward hacking via information-theoretic reward modeling. *ArXiv*, abs/2402.09345.

Chancharik Mitra, Brandon Huang, Tianning Chai, Zhiqiu Lin, Assaf Arbelle, Rogerio Feris, Leonid Karlinsky, Trevor Darrell, Deva Ramanan, and Roei Herzig. 2024. Sparse attention vectors: Generative multimodal model features are discriminative vision-language classifiers. *arXiv preprint arXiv:2412.00142*.

Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian D. Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. 2024. Rule based rewards for language model safety. *ArXiv*, abs/2411.01111.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and 1 others. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, and Others. 2022a. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 35.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022b. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2023. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. 2024. Personalizing reinforcement learning from human feedback with variational preference learning. *ArXiv*, abs/2408.10075.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Govind Ramesh, Yao Dou, and Wei Xu. 2024. GPT-4 jailbreaks itself with near-perfect success using self-explanation. *arXiv preprint arXiv:2405.13077*.

Paria Rashidinejad and Yuandong Tian. 2024. Sail into the headwind: Alignment via robust rewards and dynamic labels against reward hacking. *ArXiv*, abs/2412.09544.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Anikait Singh, Sheryl Hsu, Kyle Hsu, Eric Mitchell, Stefano Ermon, Tatsunori Hashimoto, Archit Sharma, and Chelsea Finn. 2025. Fspo: Few-shot preference optimization of synthetic preference data in llms elicits effective personalization to real users. *ArXiv*, abs/2502.19312.

Stanford Center for Research on Foundation Models (CRFM). 2023. Alpaca: A strong, replicable instruction-following model. `https://crfm.stanford.edu/2023/03/13/alpaca.html`. Accessed: 2025-05-20.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, Paul Christiano, Jan Leike, and Others. 2020a. Learning to summarize from human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 33.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020b. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Nishant Subramani, Nivedita Suresh, and Matthew 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.

Granite Vision Team, Leonid Karlinsky, Assaf Arbelle, Abraham Daniels, Ahmed Nassar, Amit Alfassi, Bo Wu, Eli Schwartz, Dhiraj Joshi, Jovana Kondic, and 1 others. 2025. Granite vision: a lightweight, open-source multimodal model for enterprise intelligence. *arXiv preprint arXiv:2502.09927*.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2024. Function vectors in large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024a. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5605–5620.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Ronald J. Williams. 2004. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Zhaofeng Wu, Michihiro Yasunaga, Andrew Cohen, Yoon Kim, Asli Celikyilmaz, and Marjan Ghazvininejad. 2025. rewordbench: Benchmarking and improving the robustness of reward models with transformed inputs. *arXiv preprint arXiv:2503.11751*.

Michihiro Yasunaga, Luke Zettlemoyer, and Marjan Ghazvininejad. 2025a. Multimodal rewardbench: Holistic evaluation of reward models for vision-language models. *arXiv preprint arXiv:2502.14191*.

Michihiro Yasunaga, Luke S. Zettlemoyer, and Marjan Ghazvininejad. 2025b. Multimodal rewardbench: Holistic evaluation of reward models for vision language models. *ArXiv*.

Kayo Yin and Jacob Steinhardt. 2025. Which attention heads matter for in-context learning? In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Chao Yu, Qixin Tan, Hong Lu, Jiaxuan Gao, Xinting Yang, Yu Wang, Yi Wu, and Eugene Vinitsky. 2024a. ICPL: Few-shot in-context preference learning via LLMs. *arXiv preprint arXiv:2410.17233*.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and 1 others. 2024b. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13807–13816.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

Hanning Zhang, Juntong Song, Juno Zhu, Yuanhao Wu, Tong Zhang, and Cheng Niu. 2025a. Rag-reward: Optimizing rag with reward modeling and rlhf. *arXiv preprint arXiv:2501.13264*.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024. Generative verifiers: Reward modeling as next-token prediction. *ArXiv*.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025b. Generative verifiers: Reward modeling as next-token prediction. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Yi-Fan Zhang, Tao Yu, Haochen Tian, Chaoyou Fu, Peiyan Li, Jianshu Zeng, Wulin Xie, Yang Shi, Huanyu Zhang, Junkang Wu, and 1 others. 2025c. Mm-rlhf: The next step forward in multimodal llm alignment. *arXiv preprint arXiv:2502.10391*.

Siyan Zhao, John Dang, and Aditya Grover. 2023. Group preference optimization: Few-shot alignment of large language models. *ArXiv*, abs/2310.11523.

Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. 2018. Interpreting deep visual representations via network dissection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 41, pages 2131–2145.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

Supplementary Material for "Activation Reward Models"

Here, we provide additional information about our experimental results, implementation details, and datasets. Specifically, Section A provides more experiment results, Section B provides additional method details, and Section C provides additional implementation details. Full data-loading code, bias-injection prompts, and inference-time prompt templates for every benchmark and baseline are released on our project page.[1]

A Additional Experiment Results

A.1 Additional Results

The results in Table 4 extend our analysis to additional models, including smaller architectures such as Granite Vision (Team et al., 2025). The ability of Activation RMs to effectively align smaller models suggests broad applicability across model scales, making the method useful for settings with constrained compute or specialized smaller models. This adaptability reinforces that leveraging internal activation patterns for preference encoding is less dependent on model size than methods requiring full-parameter finetuning.

A.2 Additional Ablations

Shots Per Sample.. In Table 5, we study the effect of the number of shots per entry used to derive activation steering vectors. For language-only tasks, increasing the shots per entry generally improves performance, as richer sets of examples yield more stable and representative activation signals for steering. This trend holds even when the underlying pool of unique preference samples is fixed (130 for LLaVA-OneVision-7B, 80 for Qwen2.5-VL-7B).

Scaling with Number of Examples.. We evaluate the effect of pool size by varying the total number of few-shot examples across 12, 20, 40, and 80. Performance scales with more examples, indicating improved steering quality, while strong performance persists across all settings, demonstrating the sample efficiency of Activation RMs.

B Additional Method Details

B.1 Attention Head Selection Details

The attention head selection component of Activation RMs automatically identifies the subset of attention heads that most effectively captures preference evaluation criteria, avoiding both the noise of using all heads and the overhead of manual selection. We adapt the REINFORCE algorithm (Williams, 2004; Hojel et al., 2025) for gradient-based optimization of this subset, as detailed in Algorithm 1.

We model head selection as learning a Bernoulli distribution over each attention head location. For a model with L layers and H heads per layer, parameters $\theta_{l,m}$ are transformed via sigmoid to produce selection probabilities. Each iteration samples 32 binary masks for variance reduction; for each mask, we extract activations only from selected heads, apply weighted PCA denoising, and inject the resulting activations during inference. The cross-entropy loss between the model output and the target token serves as a negative reward.

The REINFORCE gradient estimator is computed as $\sum_i \log p(\mathbf{b}_i) \cdot (R_i - \bar{R})$, where R_i is the normalized loss for sample i and $\bar{R}$ is the batch mean serving as a variance-reduction baseline. Optimization uses Adam with a learning rate of 0.1 for 200 iterations, with sigmoid outputs clamped to $[\epsilon, 1-\epsilon]$ ($\epsilon = 10^{-3}$) for numerical stability. We validate every 50 iterations on a held-out set and select heads with learned probabilities above $\tau = 0.5$.

In practice, this procedure identifies 10–30% of attention heads as most relevant for preference evaluation, with higher layers showing stronger selection probabilities, consistent with prior work showing that higher layers encode more task-specific information.

B.2 Flexibility to Prompt Formats

Activation RMs support few-shot reward model adaptation across multiple preference-label formats. Two prominent examples used in our experiments:

Ranked Setting.. For selecting the better of two responses: `Input: [Prompt] / Response A: [r_1] / Response B: [r_2] / Which response better meets the [specified criteria]?`

Scalar Reward Setting.. For evaluating a single response against criteria: `Input: [Prompt] / Response: [r] / Does this response meet the [specified criteria]?`

Full prompt templates for every benchmark and baseline are released on our project page.

[1] https://chancharikmitra.github.io/ActivationRM

Table 4: Evaluation of Activation RM on Language-Only RewardBench for granite-vision-3.1-2b.

Method / Model	Safety (%)	Chat (%)	Chat Hard (%)	Reasoning (%)	Overall (%)	Macro Avg. (%)
granite-vision-3.1-2b						
ZS LLM-as-a-Judge	52.13	50.44	53.07	49.50	50.71	51.28
8-shot LLM-as-a-Judge	49.02	55.26	52.76	48.89	50.02	51.48
ZS Generative Scoring	60.33	54.82	46.32	52.04	53.59	53.38
Activation RM	69.84	69.74	47.24	53.42	58.17	60.06

Table 5: Evaluation of Activation RM on RewardBench for a variety of shots per entry.

Method / Model	Safety (%)	Chat (%)	Chat Hard (%)	Reasoning (%)	Overall (%)	Macro Avg. (%)
LLaVA-OneVision-7B						
+ 2-shot	70.98	88.60	50.31	69.02	68.84	69.73
+ 4-shot	73.77	87.72	50.00	60.49	64.91	68.00
+ 8-shot	72.79	90.79	48.77	60.80	64.95	68.29
Qwen2.5-VL-7B						
+ 2-shot	78.03	91.67	57.06	76.71	75.82	75.87
+ 4-shot	75.74	93.86	54.29	77.48	75.50	75.34
+ 8-shot	73.77	87.72	50.00	60.49	64.91	68.00
+ 12-shot	76.72	92.54	61.66	75.33	75.46	76.56

Table 6: **Effect of Number of Examples on Activation RM Performance.** We evaluate Activation RMs using different numbers of calibration examples on Qwen2.5-VL with RewardBench.

Number of Examples	Safety (%)	Chat (%)	Chat Hard (%)	Reasoning (%)	Overall (%)	Macro Avg. (%)
12 examples	74.26	93.86	58.59	78.17	76.06	76.22
20 examples	77.05	94.74	56.44	78.40	76.67	76.66
40 examples	75.57	92.98	60.43	77.09	75.98	76.52
80 examples	76.39	92.98	57.06	79.25	76.88	76.42
130 examples (default)	78.03	94.74	57.06	78.86	77.24	77.17

Algorithm 1 REINFORCE-based Attention Head Selection

Require: Model F, validation set $\mathcal{V} = \{(p_i, r_i, y_i)\}_{i=1}^{v}$, learning rate α, iterations T
Ensure: Selected attention head locations λ^{ARM}

Initialize parameters $\theta_{l,m} \leftarrow -1$ for all layers l and heads m
Initialize Adam optimizer with learning rate $\alpha = 0.1$
for $t = 1$ to $T = 200$ **do**
  Compute probabilities $p_{l,m} = \text{sigmoid}(\theta_{l,m})$ clamped to $[\epsilon, 1 - \epsilon]$
  for $s = 1$ to $S = 32$ **do** ▷ Multiple samples for variance reduction
    Sample mask $\mathbf{b}_s \sim \text{Bernoulli}(p)$
    Extract activations from selected heads and apply weighted PCA
    Compute loss $\mathcal{L}_s$ = CrossEntropy(model output, target token)
    Store $\log p(\mathbf{b}_s)$ and $\mathcal{L}_s$
  end for
  Normalize losses: $R_s = (\mathcal{L}_s - \text{mean}(\mathcal{L}))/(\text{std}(\mathcal{L}) + \epsilon)$
  Compute policy gradient: $\nabla = \sum_{s=1}^{S} \log p(\mathbf{b}_s) \cdot R_s$
  Update parameters using Adam: $\theta \leftarrow \text{Adam}(\theta, \nabla)$
  if $t \mod 50 = 0$ **then**
    Validate on held-out set $\mathcal{V}$
  end if
end for
return $\lambda^{\text{ARM}} = \{(l, m) : \text{sigmoid}(\theta_{l,m}) > \tau\}$ ▷ Threshold $\tau = 0.5$

C Additional Implementation Details

For each benchmark, we designated the first 130 examples from each topical split of Multimodal RewardBench and RewardBench, and the first 80 examples from each split of PreferenceHack, as training sets for deriving Activation RM steering vectors and for few-shot baseline prompting. The remaining examples form the test set. Although these training pools define the maximum few-shot budget, Activation RM does not necessarily consume every example when generating each steering vector. Full data-loading code and inference-time prompt templates (zero-shot generative scoring, zero- and 8-shot LLM-as-a-Judge, 3-sample voting, and chain-of-thought variants) are released on our project page.

C.1 Multimodal RewardBench

Multimodal RewardBench (Yasunaga et al., 2025b) contains 5,211 expert-annotated preference triplets spanning six domains: general correctness, preference, knowledge, reasoning (math and coding), safety, and VQA. Each triplet consists of a multimodal prompt (typically an image and a textual question or instruction), a chosen response preferred by human evaluators, and a rejected response. This diversity enables fine-grained evaluation of LMMs as reward models across a wide range of multimodal preference scenarios, making it a key benchmark for assessing few-shot adaptation with Activation RMs.

C.2 PreferenceHack

PreferenceHack contains six splits targeting specific biases: three purely textual (**Length**, **Format**, **Positivity**) and three multimodal counterparts applying the same biases to captions paired with SUGARCREPE images. Each item is a paired-preference triple with a prompt and two responses, exactly one of which is faithful and unbiased while the other is engineered to exhibit the targeted bias. All synthetic biased responses were generated using gpt-4o-mini. Base datasets carry permissive licenses.

Length Bias.. Source articles are drawn from the test split of the OpenAI TLDR dataset (Stiennon et al., 2020b), a summarization corpus of approximately 3.8M Reddit posts paired with human-written summary snippets. The human snippet serves as the preferred response. gpt-4o-mini generates an intentionally verbose and repetitive summary as the biased alternative, with explicit instructions to avoid introducing new facts.

Positivity Bias.. We sample 1,000 instruction-answer pairs from the cleaned Alpaca-Instruct dataset (Stanford Center for Research on Foundation Models (CRFM), 2023).[2] The original answer serves as the preferred response, and gpt-4o-mini produces a biased rewrite that removes useful information while making the response flattering and upbeat.

Format Bias.. We draw 1,000 contexts from the test split of Anthropic's HH-RLHF dataset (Bai et al., 2022b).[3] The chosen assistant reply serves as the good response, and the rejected reply is rewritten by gpt-4o-mini into an unhelpful but plausible-sounding bulleted list. This exploits the observation by Eisenstein et al. (2023) that list-formatted responses often receive inflated scores regardless of content quality.

Multimodal Splits.. For each SUGARCREPE image (Hsieh et al., 2023), the correct caption serves as the preferred response, and a biased caption is constructed by applying one of the three textual bias manipulations to an incorrect caption (e.g., one describing the wrong color or object count). This ensures the biased response is both textually manipulated and factually incorrect with respect to the visual content, forcing the reward model to prioritize visual grounding over superficial textual cues.

In all splits we randomize which answer appears as A versus B and record the key `chosen` accordingly. The exact bias-injection prompts used for each split are provided in our released code.

C.3 RewardBench

RewardBench (Lambert et al., 2024) is a widely used benchmark for language-only reward models, consisting of prompt-chosen-rejected triplets judged by humans or advanced models (e.g., GPT-4) on helpfulness, honesty, harmlessness, coherence, and alignment with user intent. It spans three major categories: **Chat** (conversational quality and engagement), **Reasoning** (logical consistency and complex instruction following), and **Safety** (avoidance of harmful content and refusal of unsafe requests). The chosen-rejected distinctions are often subtle, requiring fine-grained understanding of language quality. As with Multimodal RewardBench, the first 130 examples of each category serve as our few-shot training set, with the remainder as the test set.

[2] https://huggingface.co/datasets/yahma/alpaca-cleaned

[3] https://huggingface.co/datasets/Anthropic/hh-rlhf