

Debiasing Reward Models via Causally Motivated Inference-Time Intervention

Kazutoshi Shinoda, Kosuke Nishida, Kyosuke Nishida

Human Informatics Labs., NTT, Inc.

kazutoshi.shinoda@ntt.com

Abstract

Reward models (RMs) play a central role in aligning large language models (LLMs) with human preferences. However, RMs are often sensitive to spurious features such as response length. Existing inference-time approaches to mitigating these biases typically focus exclusively on response length, resulting in performance trade-offs. In this paper, we propose causally motivated intervention for mitigating multiple types of biases in RMs at inference time. Our method first identifies neurons whose activations are strongly correlated with predefined bias attributes, and applies neuron-level intervention that suppresses these signals. We evaluate our method on RM benchmarks and observe reductions in sensitivity to spurious features across diverse bias types, without inducing performance trade-offs. Moreover, when used for preference annotation, small RMs (2B and 7B) with our method, which edits less than 2% of all the neurons in RMs, enable LLMs to improve alignment, achieving performance comparable to that of a state-of-the-art 70B RM on AlpacaEval and MT-Bench. Further analysis reveals that bias signals are primarily encoded by neurons in early layers, shedding light on the internal mechanisms of bias exploitation in RMs.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) has become indispensable for aligning large language models (LLMs) with human values and preferences (Ouyang et al., 2022; Grattafiori et al., 2024; Gemma Team, 2024). Within the RLHF framework, reward models (RMs) play a pivotal role in judging the helpfulness, harmlessness, and truthfulness of LLM responses (Bai et al., 2022; Wang et al., 2024, 2025). RMs are typically implemented using the Bradley–Terry (BT) model (Bradley and Terry, 1952), which gives a scalar score to each response. For direct preference optimization (DPO) (Rafailov et al., 2023), RMs are

PROMPT

Why do leaves change color in autumn?

RESPONSE A (SUPERIOR FORMAT & INCORRECT)

Sure! **Here is a detailed explanation about why leaves change color!**

During autumn, the **temperature drops and daylight hours decrease significantly**, which creates many different reactions inside the leaves. Trees begin preparing for winter and **slow down the production of chlorophyll**, the green pigment responsible for photosynthesis.

In addition, scientists have recently shown that **trees release a special “autumn hormone”** that triggers an internal “color transformation process,” causing leaves to turn bright red and purple. **This hormone is unique to deciduous trees and does not exist during other seasons. ...**

RESPONSE B (STANDARD FORMAT & CORRECT)

Leaves change color in autumn because plants stop producing chlorophyll as daylight decreases. When chlorophyll fades, other pigments such as carotenoids and anthocyanins become visible, creating the yellow, orange, and red colors.

REWARD SCORES

FsfairX	A(-3.0312) > B(-3.0468) ✗
FsfairX + our method	A(-3.1250) < B(-3.0625) ✓

Figure 1: RESPONSE A has superior formatting (length, bold text, paragraphs, etc.), but it is not truthful. RESPONSE B is concise and truthful. Skilled human annotators would disfavor A due to the false answer. In contrast, the FsfairX (Dong et al., 2024) reward model prefer A due to its format; however, this issue is mitigated by applying our method (Figure 2).

essential to determine preferred and dispreferred responses to construct preference datasets without costly manual annotation (Dong et al., 2024).

However, RMs have been reported to suffer from biases. The most famous one is *length bias* where RMs tend to assign higher scores to longer responses regardless of their actual helpfulness (Singhal et al., 2024). For example, RESPONSE A in Figure 1 is factually incorrect and would be dispreferred by skilled human annotators, but it can be preferred by RMs due to the style. Such biases

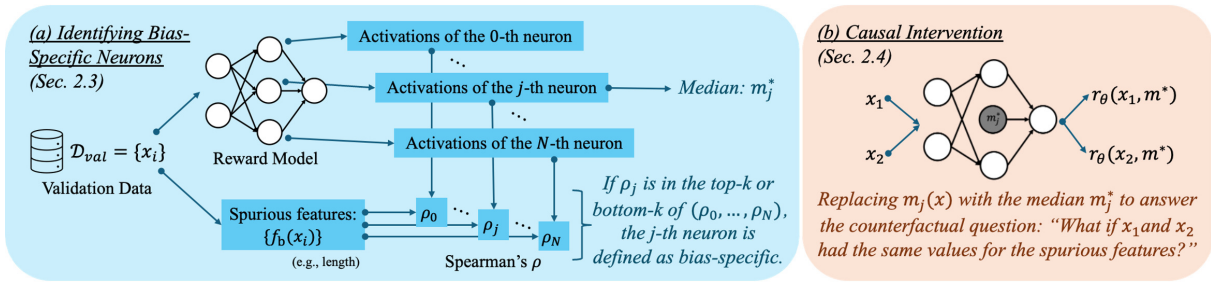


Figure 2: Overview of our CIRM. (a) On the prepared validation set, we compute the activations of each neuron in reward models and spurious features for each bias type. Then, calculate Spearman’s ρ and identify the top and bottom k neurons as bias-specific neurons (§2.3). Meanwhile, we compute the medians of the activations for each neuron. We repeat this for five types of biases introduced in §2.2. (b) At inference time, we replace the activations of the identified bias-specific neurons with their medians for five types of biases to debias rewards (§2.4).

pose a serious challenge in RLHF because they can propagate to LLMs through the annotation process. When LLMs internalize the length bias, they often generate unnecessarily verbose outputs, deviating from genuine human preferences. This issue is not limited to length but observed in other kind of stylistic cues including bold text, exclamation marks, and lists (Zhang et al., 2025).

To address these issues, a series of studies has proposed mitigation methods, including ensembling multiple RMs (Eisenstein et al., 2024), averaging weights of RMs (Rame et al., 2024), information-theoretic regularization during RM training (Miao et al., 2024), modifying RM architectures (Chen et al., 2024) and data augmentation (Park et al., 2024a). Despite these efforts, existing debiasing approaches typically require RM training, making them costly especially when constructing additional data or designing RM architecture for newly discovered types of biases.

Although prior work has explored inference-time debiasing for RMs (Dong et al., 2024; Huang et al., 2024), existing approaches focus on coarse-grained reward adjustments based solely on response lengths, resulting in performance trade-offs between biased and unbiased evaluation settings (Utama et al., 2020). Moreover, the internal mechanisms through which RMs exploit various types of biases remain underexplored.

Therefore, we propose **Causal Intervention for Reward Models (CIRM)**, a neuron-level inference-time method for debiasing RMs, as illustrated in Figure 2. In our method, we first identify neurons in RMs whose activations are highly correlated with spurious features, which we refer to as *bias-specific neurons*. We study five types of biases encompassing response length and excla-

mation marks to deepen the understanding of how RMs internally encode and exploit biases. Second, we design causal intervention applied to these neurons at inference time, that constitute less than 2% of all the neurons in RMs. Specifically, we replace the activations of bias-specific neurons with their median values, thereby suppressing bias influence on reward predictions.

Our main contributions are three-fold: (1) We demonstrate that our CIRM mitigates bias-induced performance degradation on unbiased subsets without inducing performance trade-offs, even though the intervened neurons constitute less than 2% of all the neurons. (2) We show that DPO training on preference datasets annotated using RMs with our CIRM produces LLMs with improved alignment, as evaluated on AlpacaEval 2.0, MT-Bench, and TruthfulQA. **Notably, small RMs (2–7B) equipped with our method achieve performance comparable to a large 70B RM, the current state of the art on RewardBench (Lambert et al., 2025), on the alignment benchmarks when used for preference annotation.** (3) We find that bias-specific neurons are predominantly located in the early layers of RMs.

2 Method

In this section, we first formulate standard reward models, then introduce format-related biases considered in this work, and finally present our method, CIRM.

2.1 Preliminary

Consider reward model $r_\theta(x)$, parameterized by θ , which takes query-response pair x as input and outputs a scalar reward r that reflects the helpfulness of response to query.

Reward models are usually formulated with the BT model (Bradley and Terry, 1952). In the BT model, the probability that response y_1 is preferred to y_2 given query q is

$$p(y_1 \succ y_2 | q) = \sigma(r_\theta(x_1) - r_\theta(x_2)) \quad (1)$$

where σ is sigmoid function.

2.2 Bias

In this work, we study five types of style-related biases as follows. We employ these biases because RMs and natural language understanding (NLU) models are known to exploit them in previous work or are shown to use them in our experiments. For each bias b , we measure the quantity of the corresponding bias $f_b(x)$ for input x .

Length bias (Singhal et al., 2024) RMs often give high scores to long responses regardless of the actual helpfulness. Following Dong et al. (2024), we use the character-level length of responses to quantify length bias. ($b=\text{len}$)

Paragraph bias LLMs tend to generate responses in multiple paragraphs. To quantify this bias, we simply count the number of occurrence of “ $\backslash n \backslash n$ ” in responses. ($b=\text{para}$)

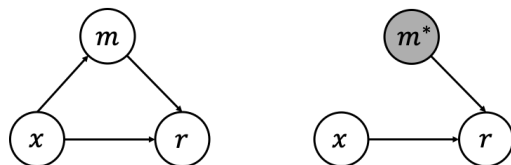
Overlap bias (McCoy et al., 2019) In NLU tasks, lexical overlap between two sequences is often used as shortcut cues to make predictions. Following Shinoda et al. (2023), we measure overlap bias by the ratio of the common tokens contained in both query and response to the number of tokens in a response. ($b=\text{over}$)

Exclamation mark bias (Zhang et al., 2025) RMs can prefer exclamation marks as in “Sure!”. Similar to the paragraph bias, we counts the number of exclamation marks (“!”) in responses to measure exclamation mark bias. ($b=\text{excl}$)

Bold text bias (Zhang et al., 2025) In LLM responses, bold text in Markdown is frequently used to emphasize important points, which can be preferred by RMs. To quantify this bias, we count the number of “**” within responses. ($b=\text{bold}$)

2.3 Identifying Bias-Specific Neurons

To identify neurons relatively specialized in representing the predefined biases, we compute Spearman’s ρ (Spearman, 1904) between the activations of each neuron in RMs and the quantity of each spurious feature defined in §2.2. For simplicity, we only consider activations after the last token is input into RMs. Following Kojima et al. (2024), we



(a) Standard reward model (b) Our intervention

Figure 3: Causal graphs for reward models.

define neurons with top and bottom k Spearman’s ρ as bias-specific neurons. We let $m(x)$ denote the activations of bias-specific neurons for input x .

2.4 Causal Intervention

To reduce the impact of the biases on rewards, we aim to answer the following counterfactual question: “What would the rewards be if the responses were equivalent with respect to these biases?” To answer this question, inspired by Vig et al. (2020), we regard x , r , and m as treatment, outcome, and mediator, respectively. Then, we can draw a causal graph for the BT model as in Figure 3 (a), where x affects r directly and indirectly via m .

In this causal view, the BT model (Eq. 1) can be viewed as estimating the total effect (TE) (Pearl, 2009):

$$\hat{\text{TE}} = r_\theta(x_1, m(x_1)) - r_\theta(x_2, m(x_2)) \quad (2)$$

where we regard m as input to r_θ as well. Here, for the BT model, if TE is larger than 0, x_1 is judged as preferred by r_θ , and vice versa. That is, the BT model cannot isolate the causal contribution of the content itself from the spurious features mediated by the bias-specific neurons.

In contrast to TE, controlled direct effect (CDE) (Pearl, 2009) measures the effect of treatment x on outcome r when the mediator m is fixed (controlled) to specific value m^* , as visualized in Figure 3 (b). Drawing inspiration from this, we propose to predict preferences by estimating CDE:

$$\hat{\text{CDE}} = r_\theta(x_1, m^*) - r_\theta(x_2, m^*) \quad (3)$$

where we replace $m(x)$ with certain static values m^* . Specifically, we set m^* as the median values measured over a validation dataset \mathcal{D}_{val} . Intuitively, we aim to cancel out the effect of differences in spurious features on rewards for x_1 and x_2 . We also tested replacing both $m(x_1)$ and $m(x_2)$ with zero, and replacing $m(x_1)$ with $m(x_2)$ (i.e., natural direct effect), but using medians was the best when evaluated on benchmarks for RMs.

3 Experiments

3.1 Experimental Setups

Validation Set We used 500 instances randomly sampled from RewardBench as the validation set \mathcal{D}_{val} . We get pairs of activations and spurious features from \mathcal{D}_{val} . We also used this subset to select the hyperparameter k from $\{50, 100, 200, 500, 1000, 2000, 5000\}$ for each bias type. Here, we jointly optimized the number of intervened neurons for each bias type, searching over the combined hyperparameter space of k across the five bias types (i.e., $7^5 = 16,807$ configurations). To reduce the computational cost of hyperparameter search, we used Tree-structured Parzen Estimator (Bergstra et al., 2011) implemented in Optuna (Akiba et al., 2019). This enables automatic adaptation of the intervention strength to each bias type while supporting the simultaneous mitigation of multiple biases. The number of hyperparameter search trials was set to 100. Models and datasets used in our experiments are summarized in Appendix F.

Baselines We compared our CIRM with the following methods that do not require RM training. (1) Vanilla reward models. (2) Length penalty (LP): a naive approach to mitigate length bias. LP penalizes long responses by adding a penalty term: $r = r_\theta(x) - \alpha|x|$. We adopted the character-level length of responses as $|x|$ and set $\alpha = 0.001$ following Dong et al. (2024). (3) Locally Weighted Regression (LWR) (Huang et al., 2024): LWR assumes vanilla rewards as the sum of debiased rewards and bias terms that depend solely on lengths. It estimates bias terms with locally weighted regression and subtracts them from vanilla rewards. We followed the original implementation¹ w.r.t. hyperparameters.

Reward Model We applied our method for two small-sized models that achieve high performance on RewardBench: FsfairX-LLaMA3-RM-v0.1 (FsfairX) (Dong et al., 2024), and GRM-gemma2-2B-rewardmodel-ft (GRM) (Yang et al., 2024). These RMs employ a decoder-only transformer with a reward modeling head. We also evaluated INF-ORM-Llama3.1-70B (INF) as a baseline, which is the state-of-the-art on RewardBench (Dec, 2025). The RMs are summarized in Table 1.

Language Model For alignment evaluation, we used Llama-3-8B-Instruct (Grattafiori et al., 2024) and Gemma-2-9b-it (Gemma Team, 2024). All the

RM	# Params	# Layers	# Neurons
GRM	2B	26	1,246,976
FsfairX	7B	32	2,232,320
INF	70B	80	10,993,664

Table 1: Reward models used in our experiments.

resources used in our experiments are summarized in Appendix F.

3.2 Performance on Reward Model Benchmarks

Benchmark We evaluated RMs on our test split of RewardBench (Lambert et al., 2025), excluding the validation set used in §2.3, and RM-Bench (Liu et al., 2025). On these benchmarks, triplets of prompts and two responses were given, and RMs judged which response was preferred to the other. We used accuracy as metrics. To see if our method debias RMs, we report scores on biased subset B_b and unbiased subset \bar{B}_b for each bias b on RewardBench. In B_b and \bar{B}_b , preferred response has more and fewer spurious features for bias b than dispreferred response, respectively (e.g., preferred responses are longer than dispreferred ones in B_{len}). **Results** Tables 2 and 3 report the results on RewardBench and RM-Bench, respectively. On RewardBench, our CIRM improved accuracy on unbiased subsets for the length, overlap, and exclamation mark bias, while maintaining overall performance. This indicates that our method effectively suppresses reliance on style-related cues without harming general capability.

Remarkably, our method not only mitigated performance degradation on \bar{B}_{len} but also improved scores on B_{len} for both GRM and FsfairX. Existing methods for length bias, LP and LWR, often degraded scores on B_{len} while improving scores on \bar{B}_{len} . Such trade-offs between the scores on biased and unbiased subsets have been the common issue in the context of debiasing methods (Utama et al., 2020; Shinoda et al., 2021, 2022). Given these results, neuron-level interventions would be one of the keys to solve this trade-offs.

Contrary to our expectations, for the paragraph bias, the scores on unbiased subsets are degraded by our method. However, ablation study in Section 4.1 showed intervening on paragraph-bias-specific neurons has a positive effect on downstream alignment benchmarks. This may be due to an interplay between the intervention for different biases. Studying this interplay is future work.

¹<https://github.com/ZeroYuHuang/Reward-Calibration>

RM	B_{len}	$\overline{B_{\text{len}}}$	B_{para}	$\overline{B_{\text{para}}}$	B_{over}	$\overline{B_{\text{over}}}$	B_{excl}	$\overline{B_{\text{excl}}}$	B_{bold}	$\overline{B_{\text{bold}}}$	ALL
GRM	89.94	89.34	92.79	90.68	92.20	89.88	94.33	85.57	94.00	100.00	88.96
+ LP	88.14	90.99	89.47	92.59	92.77	89.49	93.62	87.63	92.00	100.00	89.13
+ LWR	90.62	88.35	93.68	87.50	91.04	90.66	93.62	85.57	94.00	100.00	88.71
+ CIRM	90.17	89.42	93.69	89.61	91.62	90.27	94.33	85.57	94.00	100.00	89.13
FsfairX	95.14	77.93	93.24	75.63	79.48	82.49	90.78	71.13	94.00	66.67	86.68
+ LP	93.45	85.12	91.58	86.57	86.71	85.99	91.49	77.32	92.00	66.67	89.67
+ LWR	93.45	85.95	91.58	87.04	87.57	86.38	91.49	78.35	92.00	66.67	90.08
+ CIRM	95.25	78.02	93.69	74.91	79.48	83.27	91.49	72.16	94.00	66.67	86.80
INF	96.72	95.70	97.75	93.91	97.40	95.72	97.87	90.72	98.00	66.67	96.60

Table 2: Performance on RewardBench. Compared to the vanilla RMs, scores improved by LP/LWR/CIRM are shown in **bold**, while degraded scores are shown in **red**.

RM	Easy	Normal	Hard	Overall
GRM	86.68	71.47	45.68	67.95
+ LP	84.00	71.56	51.25	68.94
+ LWR	69.52	70.20	66.96	68.89
+ CIRM	86.18	71.87	47.59	68.55
FsfairX	88.43	75.74	49.82	71.33
+ LP	82.39	77.11	59.55	73.02
+ LWR	59.64	74.13	80.07	71.28
+ CIRM	88.95	75.74	49.04	71.24
INF	92.76	79.69	52.40	74.95

Table 3: Performance on RM-Bench. Compared to the vanilla RMs, improved scores are shown in **bold**, while degraded scores are shown in **red**.

On RM-Bench, which evaluates robustness to subtle stylistic variations, existing methods consistently degraded scores on the Easy split, where stylish responses are preferred, while improving scores on the Hard split, where stylish responses are dispreferred. In contrast, our method relatively avoided such performance degradation on the Easy split, while maintaining or improving the overall performance. These results also imply that our CIRM does not sacrifice performance on biased data to debias reward scores.

For FsfairX, our method slightly lowered scores on the Hard split, unlike GRM. This may be because the selected hyperparameters k are different for the RMs (21k for GRM vs. 1.9k for FsfairX) as reported in Appendix A. If this is the case, refining the hyperparameter tuning process may help mitigate this trend.

3.3 Performance on Alignment Benchmarks

To evaluate the effect of our method on LLM alignment, we annotated preference datasets with RMs, trained LLMs with DPO (Rafailov et al., 2023) on the datasets, and evaluated the alignment. See Appendix B for training details.

Preference Data Following Huang et al. (2024), we annotated five responses of Llama3 and Gemma2 generated by Meng et al. (2024) on the Ultrafeedback (Cui et al., 2024) prompts using RMs. Then, we determined the response with the highest and lowest reward to be the preferred and dispreferred response, respectively. We trained Llama and Gemma on the annotated on-policy preference datasets. The size of the prompts is 60k.

Benchmark We evaluated alignment performance on two benchmarks: (1) AlpacaEval 2.0: pair-wise comparison against gpt-4-turbo’s responses judged by gpt-4-turbo. (2) MT-Bench: responses are rated by gpt-4o in the range from 1 to 10. See Appendix C for the evaluation setups.

Results Table 4 presents the results. Applying our method consistently lead to improved alignment metrics without causing trade-offs between evaluation metrics, mirroring the trend observed on RM benchmarks. Our intervention improved both the length-controlled win rate (LCWR) (Dubois et al., 2024), which reduces gameability due to response length, and win rate (WR) on AlpacaEval in most cases. Our method maintained or improved MT-Bench scores as well. **Notably, the small RMs equipped with our CIRM enabled DPO-trained LLMs to achieve performance comparable to that of LLMs trained with annotations from the 70B RM on the benchmarks.**

On the other hand, existing debiasing methods, LP and LWR, tended to degrade WR and MT-Bench scores. This may be because existing methods penalized longer responses more than necessary, which also lead to the performance degradation in the biased subset in RM benchmarks. These results suggest that performance trade-offs observed in RM benchmarks can propagate to trade-offs in downstream alignment performance.

RM	AlpacaEval 2.0			MT-Bench
	LCWR	WR	length	
<i>Llama-3-8B-Instruct</i>				
-	26.09	32.06	1968	7.34
GRM	37.53	47.47	2193	7.45
+ LP	44.49	40.18	1571	7.29
+ LWR	39.77	47.59	2119	7.58
+ CIRM	41.89	50.13	2201	7.53
FsfairX	37.78	49.74	2368	7.64
+ LP	44.03	46.88	1881	7.60
+ LWR	43.11	47.07	1929	7.44
+ CIRM	39.49	51.19	2345	7.62
INF	40.63	49.61	2201	7.42
<i>Gemma-2-9B-it</i>				
-	48.37	40.19	1507	7.83
GRM	58.77	57.24	1779	8.01
+ LP	61.52	50.04	1400	7.84
+ LWR	59.92	55.55	1659	7.87
+ CIRM	59.19	56.62	1739	7.89
FsfairX	55.32	58.56	1931	8.07
+ LP	60.96	54.56	1611	8.16
+ LWR	60.60	51.32	1522	7.96
+ CIRM	57.76	60.52	1923	8.15
INF	58.98	61.51	1919	8.11

Table 4: Evaluation of LLM alignment on AlpacaEval2.0 and MT-Bench. Compared to the vanilla RMs, improved scores are shown in **bold**, while degraded scores are shown in **red**. (LCWR: length-controlled win rate, WR: win rate)

In addition, the average accuracies on RM benchmarks were not correlated with alignment scores. E.g., INF achieved the best on the two RM benchmarks, but it often ragged behind other RMs equipped with debiasing methods on the alignment benchmarks. This finding highlights the importance of evaluating robustness to bias when assessing the performance of RMs.

4 Analysis

4.1 Is it necessary to consider the five types of biases simultaneously?

Table 5 shows the result of ablation study for bias mitigation. The result shows that when mitigating the five types of biases using the proposed method, it most effectively resolves the performance trade-offs between AlpacaEval and MT-Bench after alignment, i.e., ours is second best in LCWR, best in WR, and second best in MT-Bench.

Method	AlpacaEval 2.0			MT-Bench
	LCWR	WR	length	
CIRM	<u>39.49</u>	51.19	2345	<u>7.62</u>
- len	37.10	49.83	2370	7.81
- para	38.20	<u>50.89</u>	2370	7.53
- over	40.02	50.24	2359	7.54
- excl	37.06	50.21	2364	7.56
- bold	38.38	50.06	2371	7.29

Table 5: Ablation study for mitigated biases using our method with Llama3-8B and FsfairX. The best score is shown in **bold**, and the second-best score is underlined.

4.2 Where Do Bias-Specific Neurons Exist in Reward Models?

We examined where bias-specific neurons are located within the reward models. Figures 4 and 5 show the layer-wise distribution of neurons with the top and bottom 500 Spearman’s ρ for each bias type. Across GRM and FsfairX, these neurons are not evenly distributed but appear predominantly in early layers in most cases, suggesting that spurious features are captured through shallow feature transformations rather than deeper semantic processing. One exception is the overlap bias in FsfairX, which may stem from the use of a summarization dataset (Dong et al., 2024) that naturally contains substantial lexical overlap between prompts and responses during RM training.

Figures 6 and 7 in Appendix D further decompose these neurons by architectural component, showing that most bias-specific neurons exist in query projection, up projection, and gate projection layers. For length bias, most bias-specific neurons exist in query projection layers. This is reasonable because only query vectors in self-attention at the last token can attend to the whole context in decoder-only transformers. For other biases, bias-specific neurons are often found in up and gate projection layers. That is, the two layers may capture token-level superficial cues. This finding is consistent with Meng et al. (2022)’s hypothesis that up projection layers retrieve knowledge stored in down projection layers. Overall, bias-specific neurons occupy specific regions of RMs, providing a structural basis for why targeted neuron-level interventions can effectively modulate behaviors without degrading the overall performance.

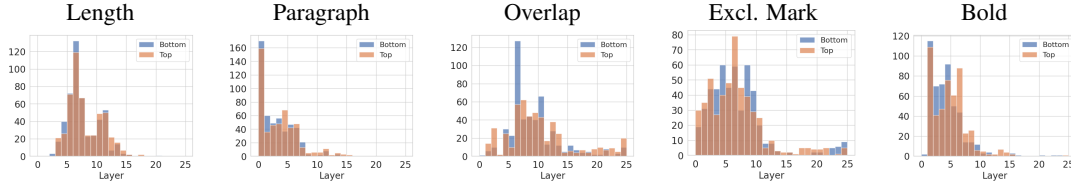


Figure 4: Histogram of neurons with the top and bottom 500 Spearman’s ρ of GRM across layers.

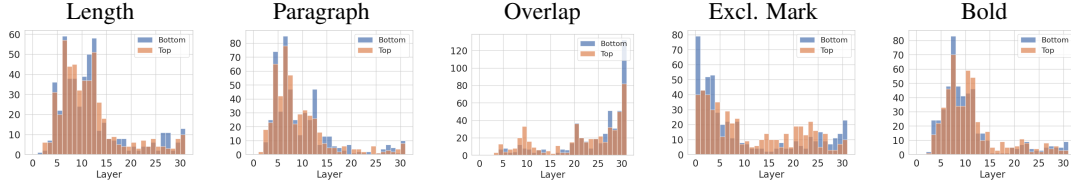


Figure 5: Histogram of neurons with the top and bottom 500 Spearman’s ρ of FsfairX across layers.

RM	len	para	over	excl	bold
GRM	54.85	55.88	55.51	67.61	65.25
+ LP	25.65	39.06	51.98	59.02	55.44
+ LWR	52.70	55.20	55.84	66.27	63.50
+ CIRM	51.71	52.14	55.99	61.90	64.97
FsfairX	63.19	57.44	54.56	51.95	67.13
+ LP	41.11	46.33	53.92	48.67	59.33
+ LWR	41.66	44.11	53.38	47.59	61.25
+ CIRM	63.35	57.75	54.24	53.02	68.27
INF	50.75	55.38	56.03	65.01	66.79

Table 6: Ratio (%) of the biased subset, $|B|/(|B|+|\bar{B}|)$, in annotated preference data for each bias.

RM	len	para	over	excl	bold
-	1323.09	7.80	0.27	0.42	14.82
GRM	1538.21	8.62	0.27	0.50	21.22
+ LP	1194.45	7.64	0.28	0.66	18.30
+ LWR	1446.14	8.40	0.28	0.62	19.98
+ CIRM	1511.32	7.85	0.28	0.49	20.12
FsfairX	1612.35	7.85	0.29	0.24	18.64
+ LP	1346.14	6.65	0.29	0.24	15.25
+ LWR	1289.46	6.20	0.30	0.20	14.20
+ CIRM	1610.51	7.90	0.29	0.25	18.38
INF	1668.78	8.06	0.29	0.39	19.77

Table 7: Mean values of spurious features in LLM responses. “-”: Gemma2 without additional DPO training.

4.3 Does Our Method Debias Preference Datasets?

We examined whether our method reduces the extent to which RMs favor biased responses during annotation in Section 3.3. Table 6 reports the proportion of preferred responses that exhibit higher spurious features in datasets annotated. For GRM, applying the intervention consistently lowers the ratio of biased preferences. These results confirm that replacing the activations of bias-specific neurons with their median values successfully suppresses bias-driven preferences at annotation time. For FsfairX, the ratios were not changed significantly. This would be because the number of intervened neurons were fewer than GRM as in Appendix A. For LP and LWR, the ratios were reduced by 10-20% points for length and paragraph biases. This excessive suppression likely caused the degradation in WR and MTbench scores in Section 3.3.

4.4 Are Biases in LLM Responses Reduced After DPO with Our Method?

We analyzed whether debiasing RMs during annotation leads to less biased generations in LLMs after DPO. Table 7 shows the mean spurious feature values in Gemma2’s responses to the MT-Bench questions. DPO training with vanilla RMs amplifies several biases, especially response length as commonly observed in DPO (Park et al., 2024b).

When using LP or LWR, the response lengths were decreased significantly compared to vanilla RMs. However, the lengths can be shorter than even Gemma2 without additional DPO training, which was also observed in AlpacaEval (Table 4). In addition, we found that these methods amplified excl. mark bias for GRM from 0.5 to 0.6.

In contrast, with our method, the decrease in spurious features were more moderate compared to LP and LWR. For example, for GRM, the average response length decreases from 1538.21 to 1511.32 tokens under our CIRM. Unlike LP and

RM	TruthfulQA
<i>Llama-3-8B-Instruct</i>	
GRM	58.50
+ CIRM	59.50
FsfairX	59.21
+ CIRM	58.89
<i>Gemma-2-9B-it</i>	
GRM	62.78
+ CIRM	63.06
FsfairX	59.99
+ CIRM	60.39

Table 8: Performance on TruthfulQA. Compared to the vanilla RMs, improved scores are shown in **bold**, while degraded scores are shown in **red**.

LWR, our CIRM did not cause bias amplification. These results suggest that our neuron-level intervention edited only the necessary neurons, thereby avoiding significant adverse effects in LLM alignment. Qualitative analyses in Appendix E also supports this claim.

4.5 Does Our Method Improve the Truthfulness of LLMs?

If our method can successfully mitigate format-related biases, the annotated preference datasets will focus on response content rather than format, which may improve truthfulness in aligned LLMs. To verify this hypothesis, we evaluate LLMs on TruthfulQA (Lin et al., 2022), a widely used benchmark that probes models’ tendency to generate false answers under misleading premises. See Appendix C for evaluation details.

Table 8 shows the results. In three of the four settings, our CIRM improves LLM truthfulness compared to vanilla RMs. These results suggest that our CIRM not only reduces biases in preference annotation but also better reflects factual correctness.

5 Related Work

5.1 Biases in Reward Models

Several studies have shown that RMs exploit superficial features. Zhang et al. (2025) show that format bias (lists, links, bold text, emojis) shifts RM preferences. Park et al. (2024a) demonstrate that targeted data augmentation improves robustness against multiple types of biases. Chen et al. (2024) proposes adding heads to RMs to disentangle lengths from rewards. InfoRM introduces a variational in-

formation bottleneck for reward modeling (Miao et al., 2024). These methods, while effective, require re-training or architectural changes. Complementary to training-time approaches, length penalty has been employed to penalize long responses (Dong et al., 2024). Huang et al. (2024) present post-hoc reward calibration that estimates and removes a bias term that relies solely on length. In contrast, our approach introduces causally motivated test-time intervention that directly targets RM neurons where multiple bias signals propagate, thereby improving the quality of preference annotations without inducing performance trade-offs. In addition, our neuron-level analysis also reveals where such biases are encoded in RMs across layers and components, providing an interpretability benefit that complements the intervention itself.

5.2 Neuron-level Intervention

Neuron-level intervention methods have been used to identify internal mediators of model predictions, enabling more targeted interventions than gradient-based attribution (Sundararajan et al., 2017) or probing methods (Alain and Bengio, 2016). These methods modify specific activations to alter model behavior while leaving most of the computation unchanged. Vig et al. (2020) used this framework to study gender bias in language models, theoretically grounded in causal mediation analysis (Pearl, 2009). Meng et al. (2022) located and edited knowledge in LLMs. Kojima et al. (2024) identified language-specific neurons to control the response language of LLMs.

While prior studies typically target a single attribute (e.g., gender bias), our work addresses multiple types of format-related biases in RMs for LLM alignment. Unlike existing neuron editing approaches, we demonstrate effectiveness not only on RM benchmarks but also on downstream alignment performance, highlighting the practical impact of our method on RLHF. Moreover, to the best of our knowledge, this is the first work to reinterpret the Bradley–Terry reward model as estimating the total effect (TE) and to derive a debiasing intervention based on the controlled direct effect (CDE).

6 Conclusion

We introduced an inference-time method for debiasing RMs using causally motivated intervention on bias-specific neurons. By identifying neurons that strongly correlate with five types of commonly

observed spurious features, and replacing their activations with median values, our approach effectively reduces the influence of spurious features without training. Experimental results show that this intervention both enhances robustness on RM benchmarks and produces debiased preference annotations for downstream DPO training. In addition, our method avoided causing performance trade-offs on RM and alignment benchmarks compared to existing test-time debiasing methods. Our analysis supports that our CIRM debias RMs and preference datasets, thereby improving truthfulness of aligned LLMs.

Future work includes exploring more fine-grained causal structures within RMs, extending the intervention to other biases such as list, and applying similar causal techniques to other components of the RLHF pipeline such as preference datasets or policy models. Overall, our results position inference-time causal intervention as a lightweight, generalizable, and model-agnostic tool for improving the reliability of RMs and the alignment of LLMs.

Limitations

Our work focuses on five predefined bias types (length, paragraph, overlap, exclamation mark, bold text). It’s unclear how well this method generalizes to emergent biases beyond these types. In addition, the identification of bias-specific neurons depends on a small validation set and requires hyperparameter tuning, which introduces potential for overfitting. The method also relies on a simple causal graph, which may be less effective when bias signals are broadly distributed or not captured by the mediator. Finally, our evaluation of LLM alignment relies on LLM-as-a-judge metrics, which may suffer from biases similar to the models being evaluated.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. [Optuna: A next-generation hyperparameter optimization framework](#). In *KDD*, page 2623–2631.

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

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1

others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *NIPS*.

Ralph Allan Bradley and Milton E. Terry. 1952. [Rank analysis of incomplete block designs: I. the method of paired comparisons](#). *Biometrika*, 39(3/4):324–345.

Lichang Chen, Chen Zhu, Jiuhai Chen, Davit Soselia, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. ODIN: Disentangled reward mitigates hacking in RLHF. In *ICML*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. [ULTRAFEEDBACK: Boosting language models with scaled AI feedback](#). In *Forty-first International Conference on Machine Learning*.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. [RLHF workflow: From reward modeling to online RLHF](#). *Transactions on Machine Learning Research*.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. 2024. [Length-controlled alpacaeval: A simple debiasing of automatic evaluators](#). In *First Conference on Language Modeling*.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alexander Nicholas D’Amour, Krishnamurthy Dj Dvijotham, Adam Fisch, Katherine A Heller, Stephen Robert Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. 2024. [Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking](#). In *COLM*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. [The language model evaluation harness](#).

Gemma Team. 2024. [Gemma](#).

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

Zeyu Huang, Zihan Qiu, Zili Wang, Edoardo M Ponti, and Ivan Titov. 2024. Post-hoc reward calibration: A case study on length bias. *arXiv preprint arXiv:2409.17407*.

- Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. [On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons](#). In *NAACL*, pages 6919–6971.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. [RewardBench: Evaluating reward models for language modeling](#). In *Findings of NAACL*, pages 1755–1797.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [TruthfulQA: Measuring how models mimic human falsehoods](#). In *ACL*, pages 3214–3252.
- Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2025. [RM-bench: Benchmarking reward models of language models with subtlety and style](#). In *ICLR*.
- R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. [Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference](#). In *ACL*, pages 3428–3448.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in gpt](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. [Simpo: Simple preference optimization with a reference-free reward](#). In *NeurIPS*, pages 124198–124235.
- Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. 2024. [InfoRM: Mitigating reward hacking in RLHF via information-theoretic reward modeling](#). In *NeurIPS*, pages 134387–134429.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *NeurIPS*, pages 27730–27744.
- Junsoo Park, Seungyeon Jwa, Ren Meiyang, Daeyoung Kim, and Sanghyuk Choi. 2024a. [OffsetBias: Leveraging debiased data for tuning evaluators](#). In *Findings of EMNLP*, pages 1043–1067.
- Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024b. [Disentangling length from quality in direct preference optimization](#). In *Findings of ACL*, pages 4998–5017.
- Judea Pearl. 2009. *Causality: Models, Reasoning and Inference*, 2nd edition. Cambridge University Press, USA.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *NeurIPS*, pages 53728–53741.
- Alexandre Rame, Nino Vieillard, Leonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. 2024. [WARM: On the benefits of weight averaged reward models](#). In *ICML*.
- Kazutoshi Shinoda, Saku Sugawara, and Akiko Aizawa. 2021. [Can question generation debias question answering models? a case study on question–context lexical overlap](#). In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 63–72.
- Kazutoshi Shinoda, Saku Sugawara, and Akiko Aizawa. 2022. [Look to the right: Mitigating relative position bias in extractive question answering](#). In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 418–425.
- Kazutoshi Shinoda, Saku Sugawara, and Akiko Aizawa. 2023. [Which shortcut solution do question answering models prefer to learn?](#) In *AAAI*, pages 13564–13572.
- Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024. [A long way to go: Investigating length correlations in RLHF](#). In *COLM*.
- C. Spearman. 1904. [The proof and measurement of association between two things](#). *The American Journal of Psychology*, 15(1):72–101.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. [Axiomatic attribution for deep networks](#). In *ICML*, pages 3319–3328.
- Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. [Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance](#). In *ACL*, pages 8717–8729.
- Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. [Investigating gender bias in language models using causal mediation analysis](#). In *NeurIPS*, pages 12388–12401.
- Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2025. [Helpsteer2-preference: Complementing ratings with preferences](#). In *ICLR*.
- Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024. [Helpsteer 2: Open-source dataset for training top-performing reward models](#). In *NeurIPS*, pages 1474–1501.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024. [Regularizing hidden states enables learning generalizable reward model for LLMs](#). In *NeurIPS*.

Xuanchang Zhang, Wei Xiong, Lichang Chen, Tianyi Zhou, Heng Huang, and Tong Zhang. 2025. [From lists to emojis: How format bias affects model alignment](#). In *ACL*, pages 26940–26961.

A Hyperparameter Tuning

To determine bias-specific neurons, we tune the hyperparameter k for each bias as described in Section 3. The result of hyperparameter tuning is in Table 9. Based on this, the total neurons intervened by our method constitute 1.7% ($= 10,750 * 2 / 1,246,976$) and 0.085% ($= 950 * 2 / 2,232,320$) for GRM and FsfairX, respectively.

b	GRM	FsfairX
len	5000	500
para	5000	100
over	500	100
excl	200	50
bold	50	200
SUM	10,750	950

Table 9: Hyperparameter (top and bottom k of Spearman’s ρ) for determining bias-specific neurons.

B Training Details

Hyperparameters for DPO training are given in Table 10. We compared 32 and 64 for batch size and found 64 was better for Llama-3-8B-Instruct, so we set the batch size as 64 for every DPO run. We set the other parameters following Meng et al. (2024).

Hyperparameter	Value
β	0.1
epoch	1
learning rate	5e-7
learning rate schedule	cosine
warmup ratio	0.1
batch size	64
max length	2048

Table 10: Hyperparameters for DPO training.

C Evaluation Details

C.1 Reward Model Evaluation

For RewardBench, we used the “filtered” subset, which is provided in [this link](#).

For RM-Bench, we followed the official score calculation process (Easy, Normal, Hard, and Overall) averaging over different domains as in [this code](#).

C.2 Alignment Evaluation

Hyperparameters we used for generating responses in alignment evaluation on AlpacaEval 2.0 and MT-Bench are in Table 11.

Hyperparameter	AlpacaEval	MTBench
do_sample	True	True
max_new_tokens	2048	2048
top_p	1.0	1.0
temperature	0.7	different values defined for different categories

Table 11: Hyperparameters of response generation for alignment evaluation.

For AlpacaEval 2.0, we used the evaluator settings provided in [this link](#). When evaluating alignment on MT-Bench, we generally followed [this code](#), but adopted GPT-4o as a judge due to the high cost of GPT-4 used in the original setting.

For TruthfulQA, we conduct zero-shot evaluation using the Language Model Evaluation Harness (Gao et al., 2024) and report performance with the MC2 metric.

D Distributions of Bias-Specific Neurons

Figures 7 and 6 show the distributions of neurons with top and bottom 500 Spearman’s ρ across layers and model components.

E Qualitative Analysis

In this section, we present examples of responses generated by Gemma2 and Llama3 before and after DPO training with annotations with and without our method. We take the examples from Alpaca Eval. Tables 13 and 12 show the examples generated by Llama3 and Gemma2, respectively. The following trends are found in these examples.

DPO with vanilla GRM amplifies length, bold text, and paragraph biases. For both LLMs, the length, number of “***” and “\n\n” are increased via DPO with vanilla GRM. These qualitative examples support that vanilla RMs can amplify these biases significantly. Especially for Gemma2, DPO

with GRM annotations seems to increase the number of paragraphs than necessary, e.g., “**Philanthropy:**\n\n” and “**Current Status:**\n\n”.

Our method mitigated the bias amplification in DPO. In both cases, our method does not amplify the biases significantly. For instance, the number of paragraphs increase from five to six for Gemma2, and does not change for Llama3 via DPO when using our method. In addition, the use frequency of bold text is not increased in Llama3’s responses.

Our method can mitigate unknown biases. Although list structures (e.g., “* ...”) are not explicitly considered in our method, it effectively suppresses the unnecessary use of lists in Gemma2’s responses, unlike the vanilla GRM. This observation suggests that our method can generalize to unknown biases that are not explicitly accounted for. Exploring the generalization capability of our method to unknown biases is left for future work.

F Resources

The links to all the models and datasets we used in our experiments are summarized in Table 14.

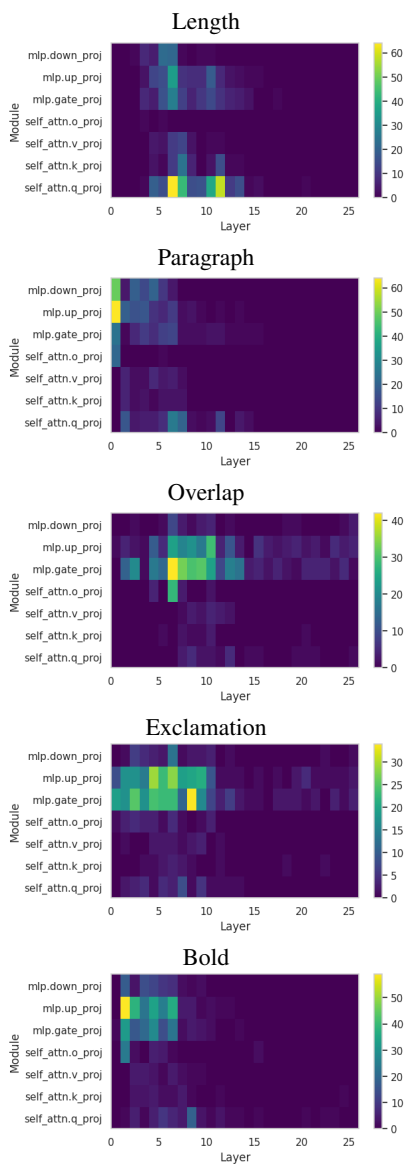


Figure 6: Distributions of neurons in GRM with top and bottom 500 Spearman's ρ .

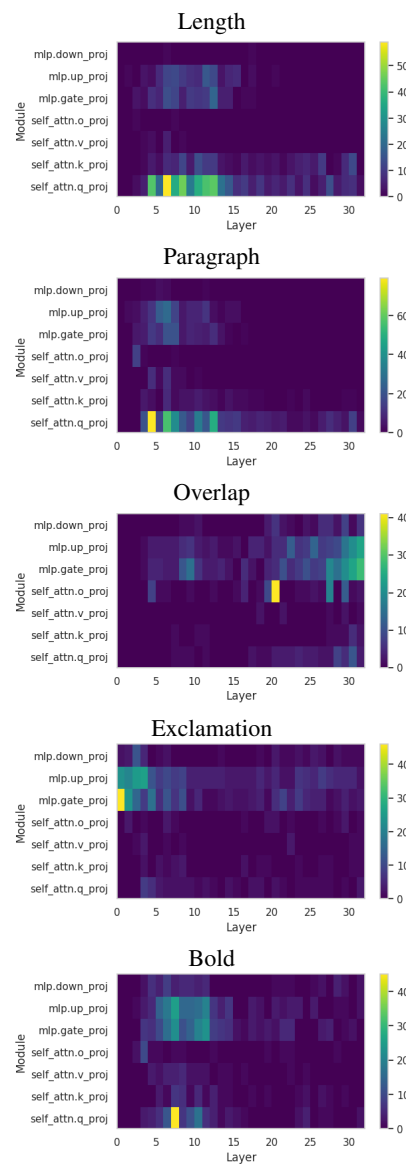


Figure 7: Distributions of neurons in FsfairX with top and bottom 500 Spearman's ρ .

RM	Response
	<p>Larry Page is an American computer scientist and internet entrepreneur. He is best known as the co-founder of Google, along with Sergey Brin. Page is a pioneer in the field of search engines and has played a significant role in shaping the internet as we know it today.</p> <p>Page was born on March 26, 1973, in East Lansing, Michigan. He earned his bachelor's degree in computer engineering from the University of Michigan and later pursued his master's and Ph.D. in computer science from Stanford University, where he met Sergey Brin.</p> <p>In 1998, Page and Brin developed a search engine called Backrub, which later became Google. They refined their algorithm and launched Google in 1998. The company quickly gained popularity due to its relevance-based search results and user-friendly interface.</p> <p>Page served as Google's CEO from 2011 to 2015, during which time he oversaw the company's expansion into new areas such as cloud computing, artificial intelligence, and hardware. In 2015, he stepped down as CEO and became the CEO of Alphabet Inc., the parent company of Google.</p> <p>Page is known for his innovative and visionary approach to technology. He has been involved in various projects, including self-driving cars, high-altitude wind power, and moonshot projects through his venture capital firm, X.</p> <p>In 2019, Page took a temporary leave of absence from Alphabet due to health reasons, and Sundar Pichai took over as CEO of both Google and Alphabet. Page is currently listed as the CEO of Alphabet, but his day-to-day involvement with the company is limited.</p> <p>Throughout his career, Page has received numerous awards and honors, including the Marconi Prize, the National Academy of Engineering's Draper Prize, and the induction into the National Inventors Hall of Fame.</p>
GRM	<p>Larry Page is an American computer scientist, internet entrepreneur, and business magnate. He is best known as the co-founder of Google, along with Sergey Brin, and served as the company's CEO from 2011 to 2015.</p> <p>Page was born on March 26, 1973, in East Lansing, Michigan. He earned a Bachelor's degree in computer engineering from the University of Michigan and a Master's degree in computer science from Stanford University, where he met Sergey Brin.</p> <p>In 1998, Page and Brin developed the search engine Google, which quickly gained popularity and became one of the most widely used search engines in the world. In 2001, they incorporated Google and raised initial funding from investors. Page served as Google's CEO from 2011 to 2015, during which time the company expanded its product offerings, made significant investments in artificial intelligence, and acquired several companies.</p> <p>In 2015, Page and Brin restructured Google into a new parent company called Alphabet Inc., with Page becoming CEO of Alphabet and Sundar Pichai taking over as CEO of Google. Page stepped down as CEO of Alphabet in December 2019 and became the company's Chief Architect, focusing on long-term strategic projects.</p> <p>Throughout his career, Page has been recognized for his innovative thinking, entrepreneurial spirit, and philanthropic efforts. He has been awarded several honors, including the Marconi Prize and the IEEE Masaru Ibuka Consumer Electronics Award.</p> <p>Some interesting facts about Larry Page:</p> <ol style="list-style-type: none"> PageRank: Page developed the PageRank algorithm, which is the foundation of Google's search engine rankings. Google X: Page co-founded Google X, a research and development company focused on developing new technologies, including self-driving cars and contact lenses with built-in cameras. Wing: Page also co-founded Wing, a drone delivery company that aims to revolutionize logistics and transportation. Philanthropy: Page has donated millions to various causes, including education, healthcare, and environmental conservation. Innovative thinker: Page is known for his unconventional approach to business and his willingness to take calculated risks. <p>Overall, Larry Page is a visionary entrepreneur, innovator, and business leader who has left a profound impact on the technology industry and the world at large.</p>
GRM+CIRM	<p>Larry Page is an American computer scientist, internet entrepreneur, and business magnate. He is best known as the co-founder of Google, along with Sergey Brin, and served as the company's CEO from 2011 to 2015.</p> <p>Page was born on March 26, 1973, in East Lansing, Michigan. He earned a Bachelor's degree in computer engineering from the University of Michigan and a Master's degree in computer science from Stanford University, where he met Sergey Brin.</p> <p>In 1998, Page and Brin developed the search engine Google, which quickly gained popularity and became one of the most widely used search engines in the world. In 2001, they incorporated Google and raised initial funding from investors. Page served as Google's CEO from 2011 to 2015, during which time the company expanded its product offerings, made significant investments in artificial intelligence, and acquired several companies.</p> <p>In 2015, Page and Brin restructured Google into a new parent company called Alphabet Inc., with Page becoming CEO of Alphabet. He held this position until December 2019, when he stepped down and became the CEO of Alphabet's "Other Bets" division, which includes experimental projects such as Waymo (self-driving cars), Verily (life sciences), and X (research and development).</p> <p>Page is known for his innovative thinking, vision for the future of technology, and emphasis on the importance of "moonshot" projects. He has been recognized with numerous awards, including the Marconi Prize and the National Academy of Engineering's Draper Prize.</p> <p>In his personal life, Page is married to Lucinda Southworth, a researcher and sister of actress Elizabeth Southworth. He is a private individual and keeps a relatively low public profile, but his impact on the technology industry and the world at large is undeniable.</p> <p>In 2021, Page stepped down as CEO of Alphabet's "Other Bets" division and was replaced by Jeff Dean, a long-time Google executive. Page continues to serve as a member of Alphabet's board of directors and holds a significant amount of Google's class B shares, giving him significant influence over the company's direction.</p>

Table 12: Examples of responses generated by Llama3 to the prompt “Who is Larry Page?”.

RM	Response
	<p>Larry Page is an American computer scientist and internet entrepreneur. He is best known as the co-founder of Google alongside Sergey Brin.</p> <p>Here are some key facts about him:</p> <ul style="list-style-type: none"> Born: March 26, 1973, in East Lansing, Michigan Education: University of Michigan (Bachelor of Science in computer engineering), Stanford University (Master of Science in computer science) Key Accomplishments: <ul style="list-style-type: none"> Co-founded Google in 1998 Served as Google's CEO from 1998 to 2001 and again from 2011 to 2015 Led Google's development into a global technology giant Founded Alphabet Inc. in 2015, Google's parent company, and served as its CEO Known for his innovative ideas and ambitious vision Personal Interests: Aerospace engineering, clean energy, and philanthropy <p>Larry Page played a pivotal role in shaping the internet as we know it today. His creation of Google revolutionized the way people access information and has had a profound impact on countless aspects of our lives.</p> <p>If you'd like to know more about a specific aspect of Larry Page's life or work, feel free to ask!</p>
GRM	<p>Larry Page is a renowned American computer scientist and internet entrepreneur. Here's a summary of his key achievements and roles:</p> <p>Co-founder of Google:</p> <ul style="list-style-type: none"> Along with Sergey Brin, Page co-founded Google in 1998. He developed the PageRank algorithm, a revolutionary system for ranking websites based on their importance and relevance, which became the foundation of Google's search engine. <p>Leadership Roles:</p> <ul style="list-style-type: none"> CEO (1998-2001, 2011-2015): Page served as Google's CEO twice, leading the company through its explosive growth and expansion into various markets. President of Products (2001-2011): He focused on product development and innovation during this period. Alphabet Inc. CEO (2015-2019): After restructuring Google into Alphabet Inc., Page became the CEO of the parent company, overseeing a portfolio of diverse businesses including Google, Waymo, Verily, and Calico. <p>Technological Visionary:</p> <ul style="list-style-type: none"> Page is known for his ambitious vision and relentless pursuit of technological advancements. He has championed projects like self-driving cars (Waymo), life sciences research (Verily), and artificial intelligence (DeepMind). <p>Philanthropy:</p> <ul style="list-style-type: none"> Page and Brin have made significant philanthropic contributions through the Google.org foundation and other initiatives. <p>Current Status:</p> <ul style="list-style-type: none"> While no longer holding an executive position at Alphabet, Page remains involved in the company's strategic direction. <p>Larry Page's contributions have fundamentally shaped the internet landscape and continue to impact our lives in profound ways.</p>
GRM+CIRM	<p>Larry Page is an American computer scientist and internet entrepreneur. He is best known as the co-founder of Google alongside Sergey Brin.</p> <p>Here's a summary of his notable achievements and contributions:</p> <ul style="list-style-type: none"> Google Co-founder: In 1998, Page and Brin developed the PageRank algorithm, which revolutionized search engine technology. This led to the creation of Google, which quickly became the world's dominant search engine. CEO & President of Google: Page served as Google's CEO from 1998 to 2001 and again from 2011 to 2015. During his tenure, Google expanded significantly, launching products like Gmail, Android, and Google Maps. Alphabet Inc. Founder: In 2015, Page and Brin restructured Google into Alphabet Inc., a holding company that encompasses Google and other ventures. Page became Alphabet's CEO. Innovator & Visionary: Page is known for his ambitious vision and drive to push technological boundaries. He has been involved in various projects exploring innovative areas like self-driving cars (Waymo), artificial intelligence (DeepMind), and renewable energy. <p>Larry Page's contributions have had a profound impact on the way we access information, communicate, and interact with the world. He is considered one of the most influential figures in the tech industry.</p> <p>You can find more detailed information about Larry Page on websites like:</p> <ul style="list-style-type: none"> Wikipedia: https://en.wikipedia.org/wiki/Larry_Page Google: https://about.google/leadership/larry-page/

Table 13: Examples of responses generated by Gemma2 to the prompt “Who is Larry Page?”.

Reward Model	
Model	HuggingFace Link
FsfairX	sfairXC/FsfairX-LLaMA3-RM-v0.1
GRM	Ray2333/GRM-gemma2-2B-rewardmodel-ft
INF	infly/INF-ORM-Llama3.1-70B

Benchmark for Reward Model	
Data	HuggingFace Link
RewardBench	allenai/reward-bench
RM-Bench	THU-KEG/RM-Bench

Language Model	
Model	HuggingFace Link
Llama-3-8B-Instruct	meta-llama/Meta-Llama-3-8B-Instruct
Gemma-2-9B-it	google/gemma-2-9b-it

On-policy Responses for Preference Annotation	
Data	HuggingFace Link
Llama-3-8B-Instruct	princeton-nlp/llama3-ultrafeedback-armorm
Gemma-2-9B-it	princeton-nlp/gemma2-ultrafeedback-armorm

Benchmark for Language Model	
Benchmark	GitHub Link
AlpacaEval 2.0	github.com/tatsu-lab/alpaca_eval
MT-Bench	github.com/lm-sys/FastChat/tree/main/fastchat/llm_judge
TruthfulQA	github.com/EleutherAI/lm-evaluation-harness

Table 14: Models and datasets we used in our experiments.