Post-Hoc Reward Calibration: A Case Study on Length Bias

Zeyu Huang [1] Zihan Qiu [2] Zili Wang [3]
Edoardo M. Ponti [1] Ivan Titov [1,4]

1 THE UNIVERSITY of EDINBURGH 2 Qwen 3

Presenter : Zeyu Huang
E-mail : zeyu.huang@ed.ac.uk

Code : https ://github.com/ZeroYuHuang/Reward-Calibration

ICLR 2025

Content

RLHF and Reward Model

RLHF has become an important technique that drives the success of LLMs. A key component in this process is the reward model (**RM**), which translates human feedback into a training signal.

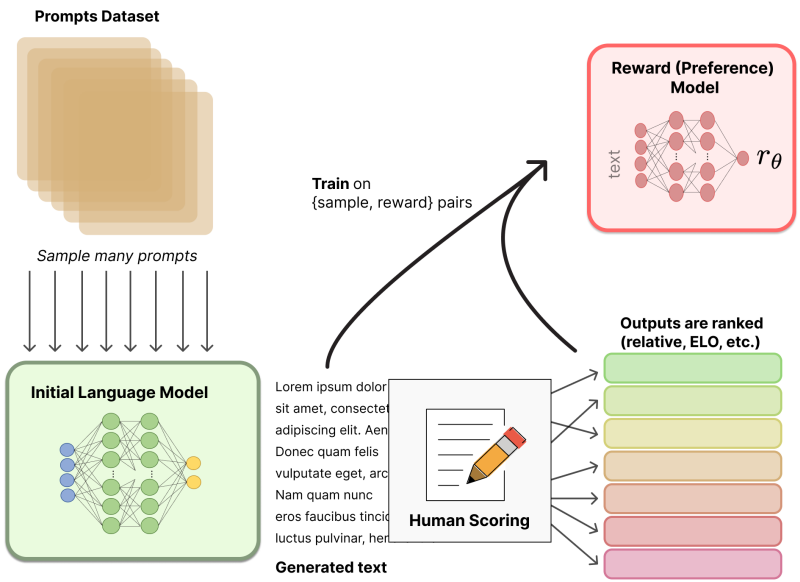


FIGURE – The reward model is usually trained on human-annotated preference pairs to predict the preference probability $p(y_1 > y_2|x)$, where y_1 and y_2 are two responses for the same prompt x.

Reward Hacking

However, a trained RM may be biased (e.g., favouring outputs based on length or style). Using a biased RM can have significant drawbacks.

1. Problematic/hackable evaluation.
2. Amplified Biases after the RLHF.

Recent advancements in alleviating reward hacking usually require extra training or data annotation :

1. Data Handling : collect/create unbiased data for the target bias
2. RM Engineering : Ensemble, Model merging, Disentangled training
3. Modifying Reinforcement Learning Algorithms : Penalty term

Research Question

We ask, can we correct or mitigate biases in reward signals without extra training and data?

This paper attempts to answer this question by framing it as *Post-hoc Reward Calibration*.

Specifically, our method uses only a batch of scored prompt-response examples to mitigate the RM's bias *without* intervening in the preference data collection, RM training, and the RLHF phase.

Post-hoc Reward Calibration : Problem Statement

Given an input–output pair x, a biased reward $r_\theta(x)$ from RM could be decomposed into two terms, an underlying calibrated reward $r^*_\theta(x)$ and a bias $b^\theta_c(c(x))$ with regard to the characteristic $c(x)$:

$$r_\theta(x) = r^*_\theta(x) + b^\theta_c(c(x)) \tag{1}$$

The RM is usually used to calculate the margin between two outputs (x_1, x_2) to predict the human preference :

$$\Delta_{r_\theta}(x_1, x_2) = r_\theta(x_1) - r_\theta(x_2) = r^*_\theta(x_1) - r^*_\theta(x_2) + \underbrace{b^\theta_c(c(x_1)) - b^\theta_c(c(x_2))}_{\text{Difference owing to bias}} \tag{2}$$

Thus, the reward calibration problem is to estimate the reward difference owing to the bias term and subtract it to recover the underlying true reward margin $\Delta^*_{r_\theta}(x_1, x_2) = r^*_\theta(x_1) - r^*_\theta(x_2)$.

Post-hoc Reward Calibration : Assumptions

1. **Independence of the biased characteristic** : consider a general reward modelling dataset, the expectation of the underlying gold reward margin should be zero and is independent of the biased characteristic c ;

2. **Lipschitz Continuity** : the bias term b_c^θ is a *slow-varying function* of characteristic c, which means if pairs x_1 and x_2 are close to each other regarding the characteristic c, their corresponding bias term in should be close as well.

Post-hoc Reward Calibration : Bias Estimation

Intuitively, to estimate $b_c^\theta(c(x_1)) - b_c^\theta(c(x_2))$, we ask : given the characteristic measures $c(x_1)$ and $c(x_2)$ for an arbitrary pair (x_1, x_2), what would be their reward margin ?

A straightforward estimation for this question could be

$$\mathbb{E}[r_\theta(x) \mid c(x) = c(x_1)] - \mathbb{E}[r_\theta(x) \mid c(x) = c(x_2)]. \tag{3}$$

Therefore, the question now is how to calculate this conditional expectation.

We propose a naive Uniform Averaging and Locally Weighted Regression to do so. For Uniform Average, we define a neighbourhood with d and calculate the local average :

$$\mathbb{E}[r_\theta(x) \mid |c(x) - c(x_1)| < d] - \mathbb{E}[r_\theta(x) \mid |c(x) - c(x_2)| < d] \tag{4}$$

d is a threshold distance that governs the neighbourhood size around x_1 and x_2.

Locally Weighted Regression (LWR)

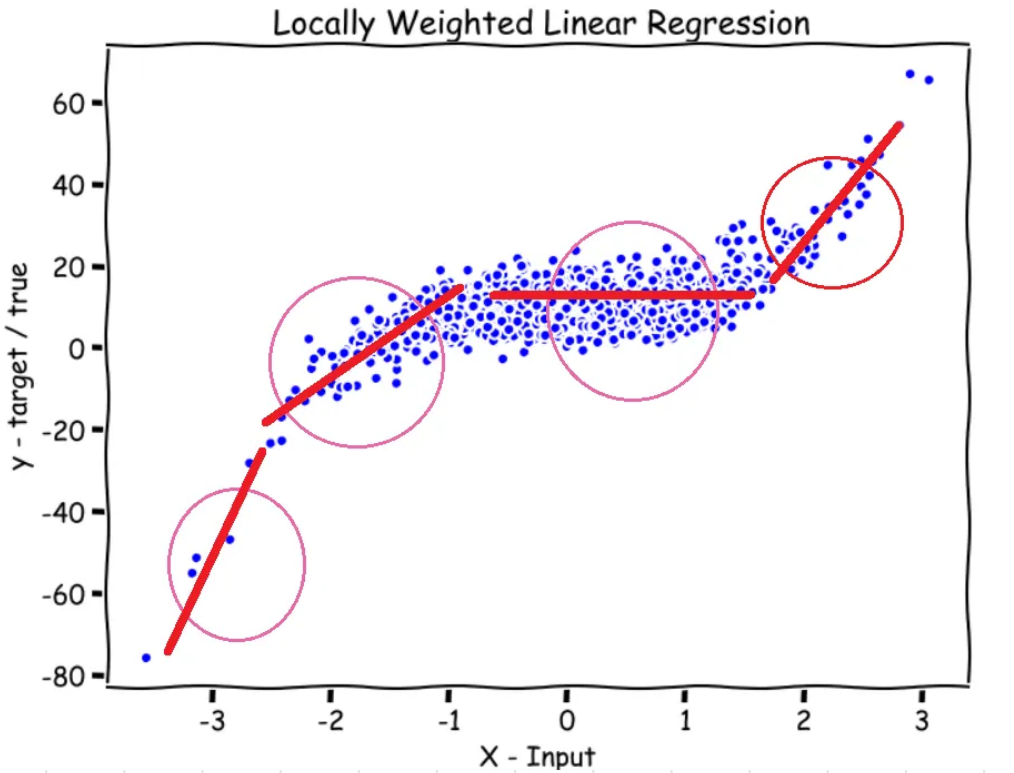
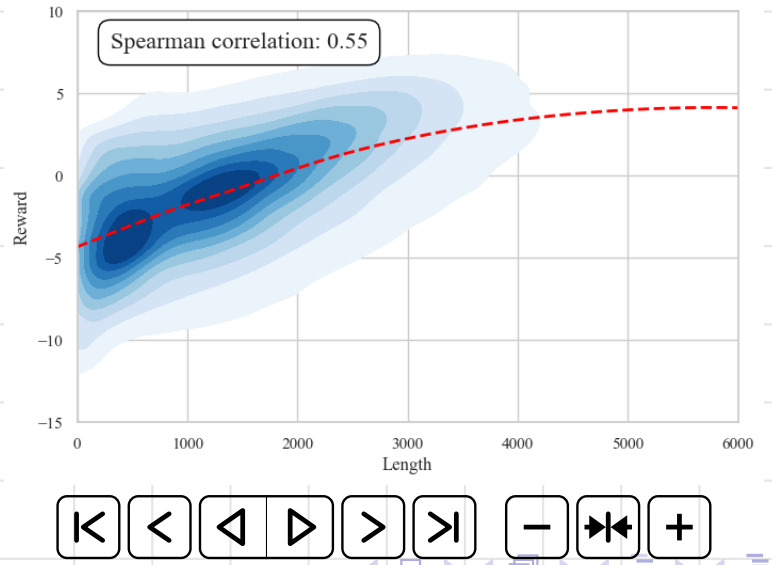


FIGURE – Locally Weighed Regression : Assign weights to data points in the neighbourhood and do weighted linear regression to estimate the local average.

Takeaway - 1 :Improved performance

Focusing on the prevalent length bias, our proposed method :

1. a **3.11** performance gain averaged over **33** RMs on the RewardBench.
2. Based on AlpacaEval, we utilise **8** open-source RMs to rank **184** LLMs. After our calibration, the rankings correlate better with GPT-4 evaluations and human preferences.
3. We assess calibrated rewards for RLHF process. Testing with four LLM–RM configurations, we observe consistent improvements in AlpacaEval2 Length-controlled win rates.
4. The calibrated rewards exhibit weak correlations with the length.



Takeaway - 2 : Generalization, Data & Computational Efficiency

1. The method requires no additional data annotation or RM retraining and is computationally efficient, e.g., calibrating over 300k samples takes less than 1 minute with a single CPU.
2. The method generalises to other quantifiable biases, and to pairwise GPT4-as-Judge models.
3. For non-biased RMs, our method minimally alters the reward behaviour, there is no harm to use it!
4. The method is robust to hyperparameter choices and dataset size.

Settings and Baselines

Settings :

1. Length Calibrated Rewards on RewardBench
2. Length Calibrated Rewards as LLMs Evaluators
3. Length Calibrated Rewards for LLM alignment

Baselines :

1. Original Reward : $r_\theta(x)$
2. Length penalty : $\hat{r^*}(x) = r_\theta(x) - \alpha \times |x|$
3. RC-Mean : the uniform average approach
4. RC-LWR : the Locally Weighted Regression Approach
5. RC-LWR-Penalty : our proposed method combined with the length penalty

Length Calibrated Rewards on RewardBench

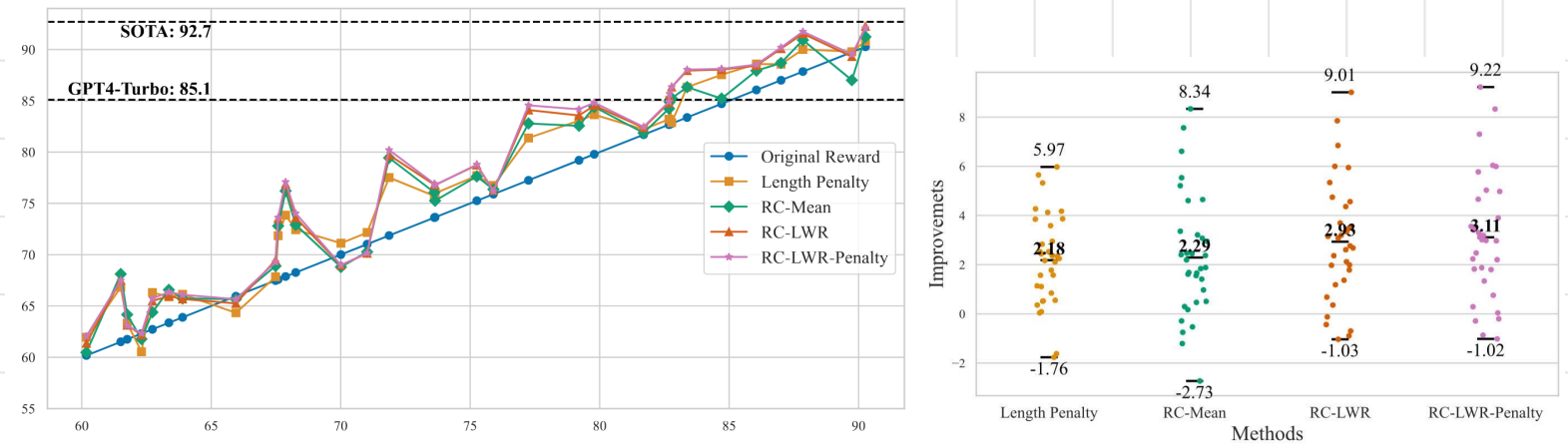
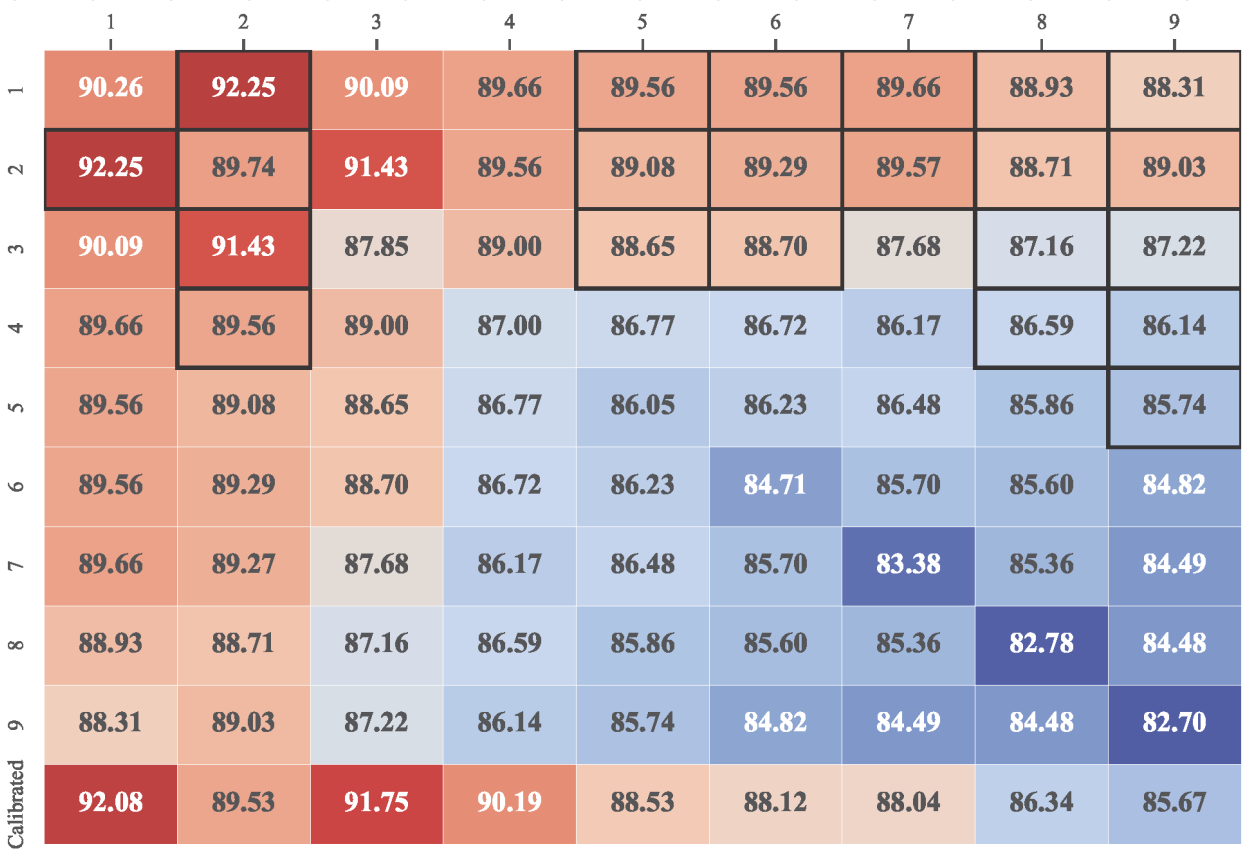


FIGURE – Results of different calibration algorithms on the RewardBench benchmark for 33 reward models are shown in two charts. **Left** : the line chart demonstrates the RewardBench score before calibration (x-axis) and after calibration (y-axis) of different algorithms for different models. **Right** : the scatter plot highlights the performance gains achieved by different calibration methods, annotated with the maximum, average, and minimum values. Our method performs best and nearly improve all reward models.

Reward Calibration and Reward Ensemble



	1	2	3	4	5	6	7	8	9
1	90.26	92.25	90.09	89.66	89.56	89.56	89.66	88.93	88.31
2	92.25	89.74	91.43	89.56	89.08	89.29	89.57	88.71	89.03
3	90.09	91.43	87.85	89.00	88.65	88.70	87.68	87.16	87.22
4	89.66	89.56	89.00	87.00	86.77	86.72	86.17	86.59	86.14
5	89.56	89.08	88.65	86.77	86.05	86.23	86.48	85.86	85.74
6	89.56	89.29	88.70	86.72	86.23	84.71	85.70	85.60	84.82
7	89.66	89.27	87.68	86.17	86.48	85.70	83.38	85.36	84.49
8	88.93	88.71	87.16	86.59	85.86	85.60	85.36	82.78	84.48
9	88.31	89.03	87.22	86.14	85.74	84.82	84.49	84.48	82.70
Calibrated	92.08	89.53	91.75	90.19	88.53	88.12	88.04	86.34	85.67

FIGURE – Ensemble performance and calibrated performance. The cell $[i, j]$ denotes the ensemble performance of model i and model j where i and j are the model's rank on the RewardBench leaderboard. The last row is the calibrated performance of models. The calibration outperforms most ensemble results except for ensembles with a much stronger model. And intuitively, ensemble-based methods may not address common bias across models.

Length Calibrated Rewards as LLMs Evaluators

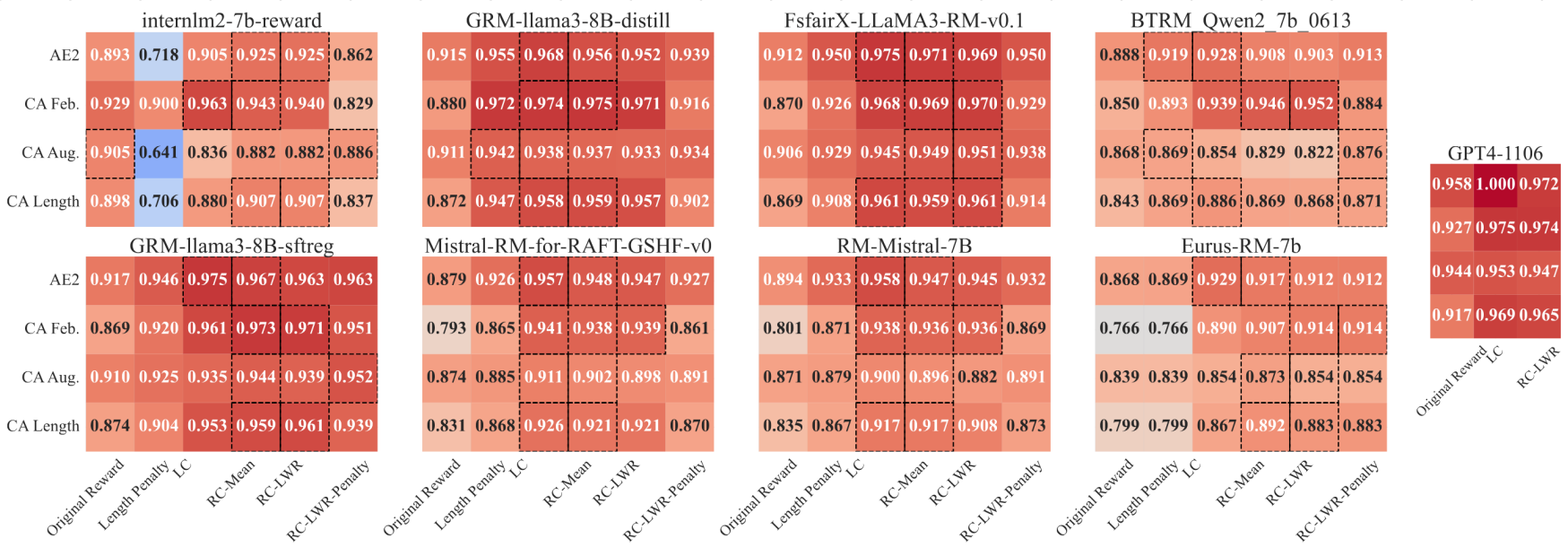


FIGURE – The heatmap demonstrates the enhanced Spearman correlation (↑) between reward-models-produced rankings with the AlpacaEval2 (AE2) and ChatbotArena (CA) after calibration. After calibration, the reward model could produce more reliable evaluations. And our method also works for the GPT4-as-judge model.

Length Calibrated Rewards for RLHF

Algorithm	Length-Controlled Win Rate				Avg Performance			
	no DPO	OR	LP	RC LWR	no DPO	OR	LP	RC LWR
Llama-3-8B-Fsfairx	23.91	41.82	49.98	**51.37**	62.69	60.45	61.75	**61.94**
Llama-3-8B-GRM	23.91	41.00	48.41	**50.49**	62.69	60.94	**62.11**	61.83
gemma2-9b-Fsfairx	46.96	62.79	67.46	**69.54**	63.69	58.88	60.52	**60.79**
gemma2-9b-Grm	46.96	63.21	66.28	**70.45**	63.69	55.28	57.28	**58.31**

TABLE – Results of using length-calibrated rewards for LLMs' alignment. We report the Length-Controlled win rate and average performance on eight benchmarks. The Table illustrates that RC-LWR calibration achieves up to 10% LC win rate improvement and alleviates the performance drop on benchmarks.

Generalize to other bias types : Markdown Features

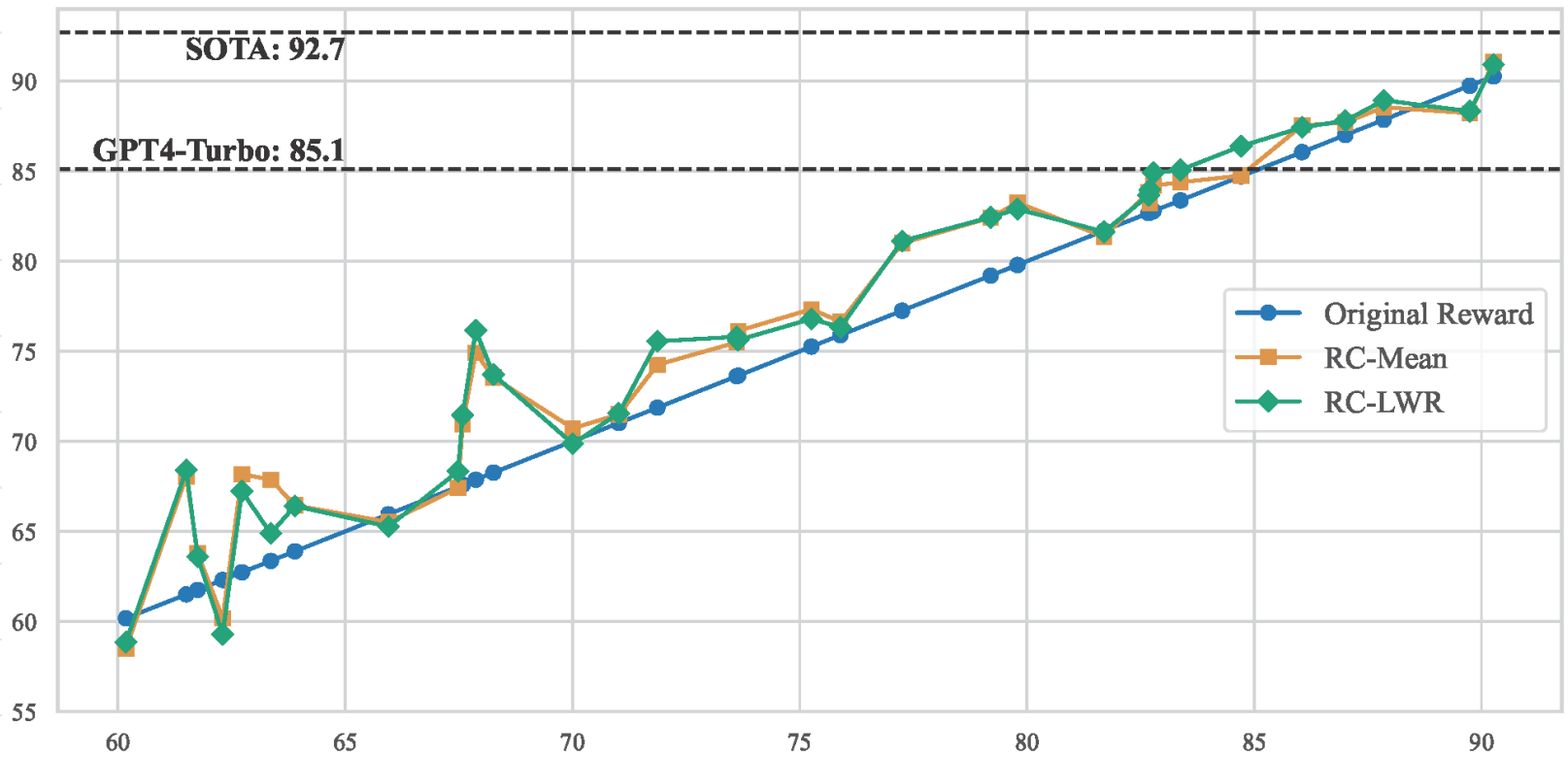


FIGURE – RewardBench score before (x-axis) and after (y-axis) markdown-feature-calibration.

Calibration behaviours on RMs with different length preferences.

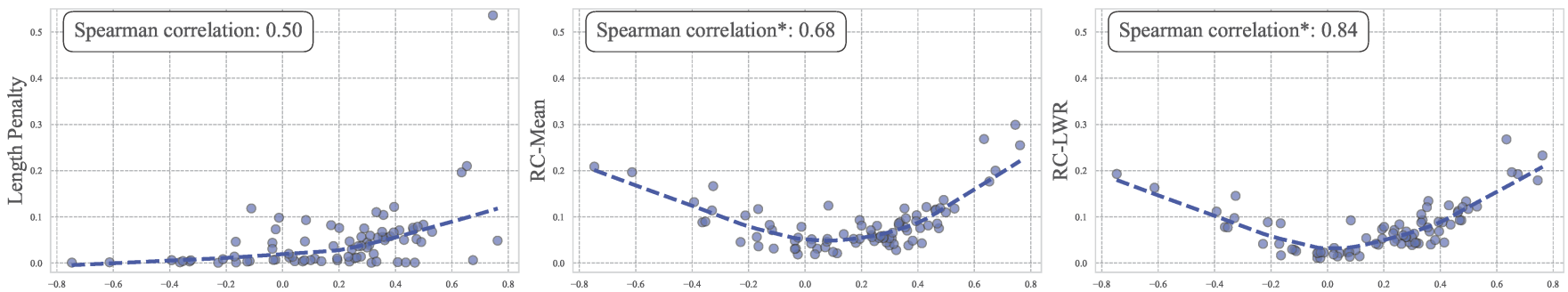


FIGURE – The number of preferences reversed by the calibration (y-axis) for RMs with different length preferences (x-axis), measured by the Spearman correlation with length. The calibration impact of the proposed method is proportional to the bias level of the reward model..

Controlling Calibration Effects

One critical assumption for our proposed method is that the underlying gold reward is independent of the biased characteristic.

However, this assumption may be invalid if one focuses on a specific subset of instructions.

A more practical calibration method should be controllable for users. We show that our proposed method can be controlled by introducing a calibration constant γ by :

$$\hat{\Delta}^{*}_{r_\theta}(x_1, x_2) = \Delta_{r_\theta}(x_1, x_2) - \underbrace{\gamma}_{\text{Calibration Constant}} \times \underbrace{(\hat{r}_\theta(c(x_1)) - \hat{r}_\theta(c(x_2)))}_{\text{Bias Term Estimated}}$$

Controlling Calibration Effects

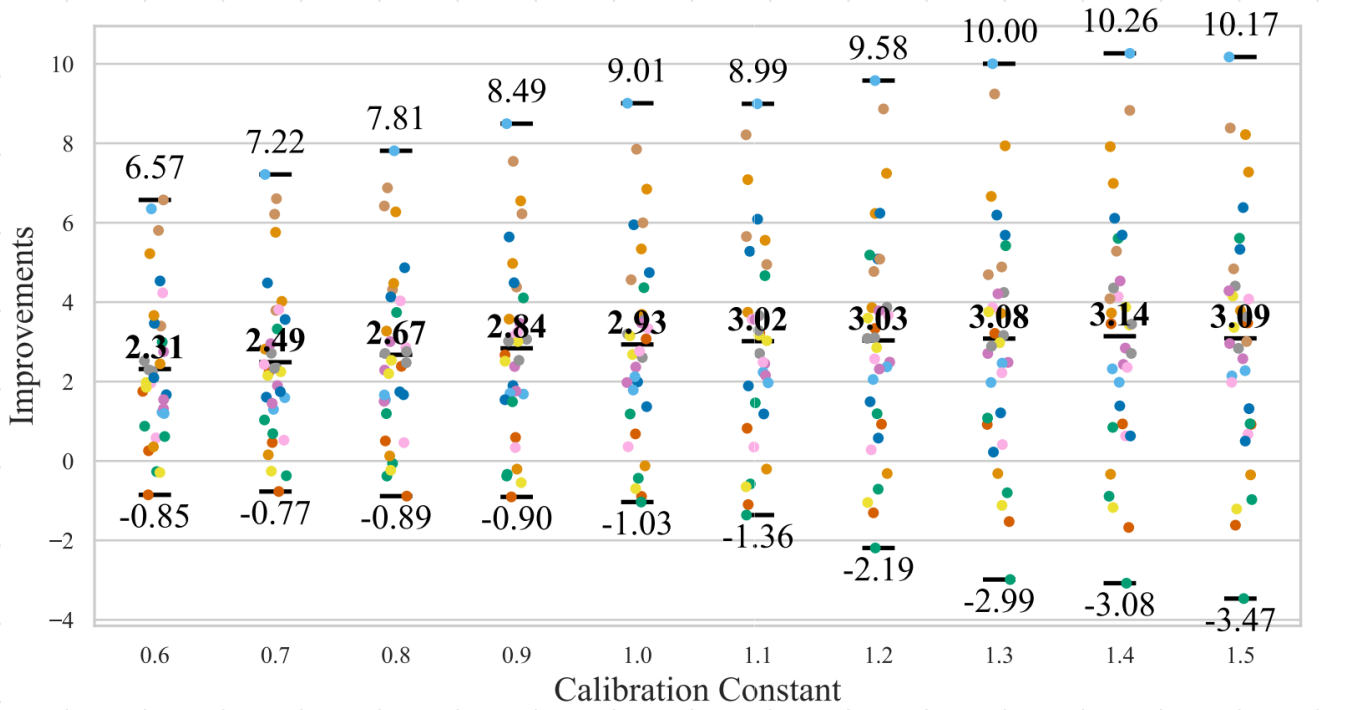


FIGURE – Ablation results of different calibration constants on the RewardBench for BT-based RMs, showing that the calibration constant can smoothly control the calibration effect. This significantly enhances the practical utility of the proposed method, as one can adjust γ to achieve superior rewarding performance if certain prior knowledge about the correlation between rewards and the characteristic of interest is known.

Thanks for your attention!!

E-mail : zeyu.huang@ed.ac.uk
Code : https ://github.com/ZeroYuHuang/Reward-Calibration

Paper : https ://openreview.net/pdf?id=Iu8RytBaji