Google DeepMind

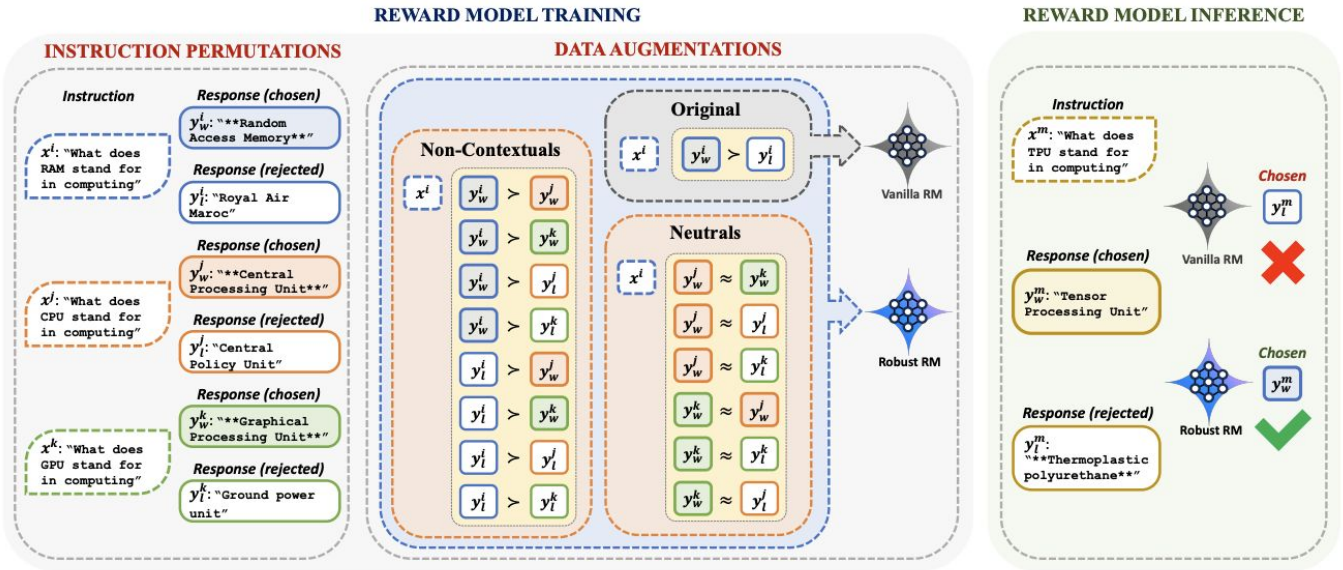
RRM: Robust Reward Model Training Mitigates Reward Hacking

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, Mohammad Saleh

Google DeepMind

Existing Issues with Traditional Reward Model Training

Response pairs are always tied to specific prompt. Thus the RM struggles to disentangle prompt-driven preferences from prompt-independent artifacts (length, format, markdown, emojis).



A Causal Perspective

Google DeepMind

Two hypothesis:

- H_0: there is no causal edge from A to C
- H_1: there is a causal edge from A to C

Proposition 3.1. *In traditional reward model training, $\mathcal{H}_0$ and $\mathcal{H}_1$ are not always distinguishable.*

- **R1**: Under $\mathcal{H}_0$, A and C are d-separated by (Y_1, Y_2), thus $A \perp C \mid (Y_1, Y_2)$.
- **R2**: Under $\mathcal{H}_0$, A and C are d-separated by S, thus $A \perp C \mid S$.

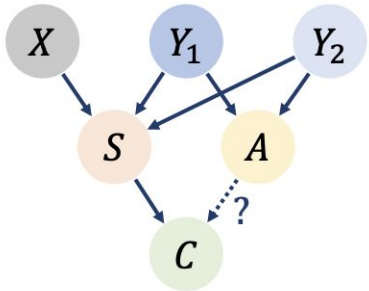


Figure 2: Causal graph of reward model. X is the prompt, Y_1, Y_2 are two responses, S is the contextual signal that depends on input prompt and two responses. A is the context-free artifact that only depends on two responses. C is the preference label. Traditional reward model cannot differentiate the two DAGs on whether there is a causal edge from A to C. Our work uses the augmented dataset to eliminate the edge from A to C.

Data Augmentation Approach

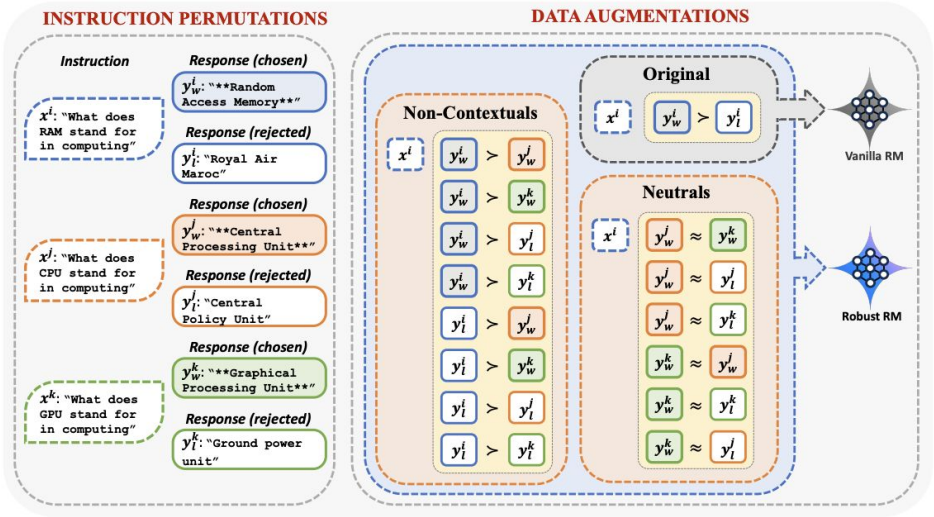
Google DeepMind

- **R1**: Under $\mathcal{H}_0$, A and C are d-separated by (Y_1, Y_2), thus $A \perp C \mid (Y_1, Y_2)$.
- **R2**: Under $\mathcal{H}_0$, A and C are d-separated by S, thus $A \perp C \mid S$.

Proposition 3.2. *If the reward model is trained with $\mathcal{D}_{hf}$ and augmented triplets in Equation 5, there is no causal edge from A to C in DAG $\mathcal{G}$.*

$$\left.\begin{array}{l}
(x^{(i)}, y_w^{(i)}, y_w^{(\sigma_1(i))}) \to \text{chosen} = y_w^{(i)} \\
(x^{(i)}, y_w^{(i)}, y_w^{(\sigma_2(i))}) \to \text{chosen} = y_w^{(i)} \\
(x^{(i)}, y_w^{(i)}, y_l^{(\sigma_1(i))}) \to \text{chosen} = y_w^{(i)} \\
(x^{(i)}, y_w^{(i)}, y_l^{(\sigma_2(i))}) \to \text{chosen} = y_w^{(i)} \\
(x^{(i)}, y_l^{(i)}, y_w^{(\sigma_1(i))}) \to \text{chosen} = y_l^{(i)} \\
(x^{(i)}, y_l^{(i)}, y_w^{(\sigma_2(i))}) \to \text{chosen} = y_l^{(i)} \\
(x^{(i)}, y_l^{(i)}, y_l^{(\sigma_1(i))}) \to \text{chosen} = y_l^{(i)} \\
(x^{(i)}, y_l^{(i)}, y_l^{(\sigma_2(i))}) \to \text{chosen} = y_l^{(i)}
\end{array}\right\} \text{Non-contextuals}$$

$$\left.\begin{array}{l}
(x^{(i)}, y_w^{(\sigma_1(i))}, y_l^{(\sigma_1(i))}) \to \text{Tie} \\
(x^{(i)}, y_w^{(\sigma_2(i))}, y_l^{(\sigma_2(i))}) \to \text{Tie} \\
(x^{(i)}, y_w^{(\sigma_1(i))}, y_w^{(\sigma_2(i))}) \to \text{Tie} \\
(x^{(i)}, y_w^{(\sigma_1(i))}, y_l^{(\sigma_2(i))}) \to \text{Tie} \\
(x^{(i)}, y_w^{(\sigma_2(i))}, y_l^{(\sigma_1(i))}) \to \text{Tie} \\
(x^{(i)}, y_l^{(\sigma_1(i))}, y_l^{(\sigma_2(i))}) \to \text{Tie}
\end{array}\right\} \text{Neutrals}$$



Connection to Existing Works

Google DeepMind

Existing Work	Key Ideas	Delta Value of Our Work
ODIN (Chen et al.)	disentangles length from quality during RM training and only use quality head for RL	Our work is more general and can cao beyond single and observed artifact
Length-controlled AlpacaEval-2 (Dubois et al., 2024a)	learns the residual win-rate by controlling length through Controlled Direct Effect (VanderWeele, 2011)	We directly learn the residual part that is orthogonal to the artifacts and go beyond single and observed artifact
Length-controlled DPO (Park et al., 2024)	adds a length penalty in the RLHF objective	We directly learn a artifact-free reward model so we do not need an explicit length adjustment factor in the alignment algorithm designs.
Contrast Instructions (Shen et al., 2023a)	proposes a data augmentation training approach and retrieval-augmented inference technique	We consider all possible combinations of (prompt, response 1, response 2) across different examples with organic data

Experiments - Main Results

Google DeepMind

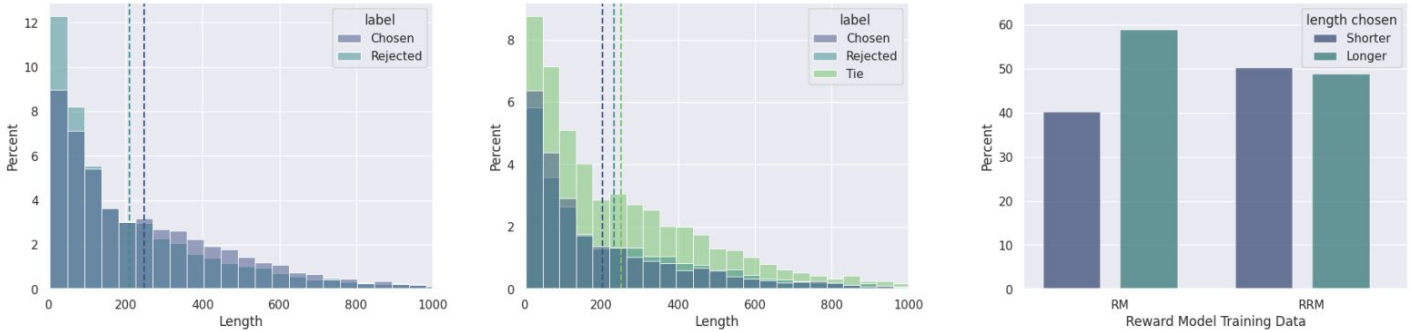
Model	Chat	Chat Hard	Safety	Reasoning	Average
RM	**97.77**	51.54	78.54	**94.58**	80.61
RRM	96.51	**65.57**	**83.90**	90.62	**84.15**

Table 1: Comparison of test accuracy of Reward-Bench. RRM shows improvement upon RM on Chat Hard and Safety with an average 3.54% improvement of accuracy.

Reward	Policy	MT-Bench[7]			AlpacaEval-2		
		T1 (↑)	T2 (↑)	Overall (↑)	LC (%) (↑)	WR (%) (↑)	Length (↓)
RM	BoN (N=8)	-	-	-	36.87	50.14	3072
RRM	BoN (N=8)	-	-	-	**47.68**	**53.19**	**1770**
RM	BoN (N=64)	-	-	-	40.52	57.62	2992
RRM	BoN (N=64)	-	-	-	**62.82**	**63.03**	**1770**
RM	DPO	8.02	6.33	7.27	33.46	41.07	2416
ODIN	DPO	8.66	8.13	8.39	48.29	37.13	**1559**
RRM	DPO	**8.70**	7.87	8.31	**52.49**	**43.31**	1723
-Neutrals	DPO	8.65	**8.21**	**8.44**	51.73	43.24	1722

Table 2: Comparison among different reward models on various aligned policies. T1 and T2 stand for the first and second turn of the conversation, respectively. WR stands for win-rate against GPT-4. LC stands for length-controlled win-rate. Length is the average number of characters in the generated responses. RRM shows quality improvements over ODIN and RM with shorter responses than RM. Dropping augmented neutral examples slightly hurt the quality.

Length Analysis - Training Data



(a) Histogram of response lengths in RM training data.

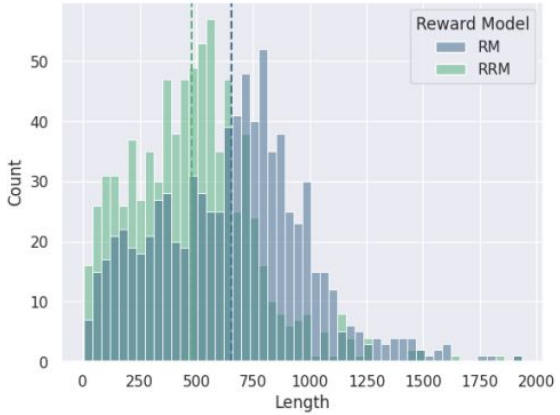
(b) Histogram of response lengths in RRM training data.

(c) Percentage of chosen responses being longer or shorter in RM and RRM traininng data.

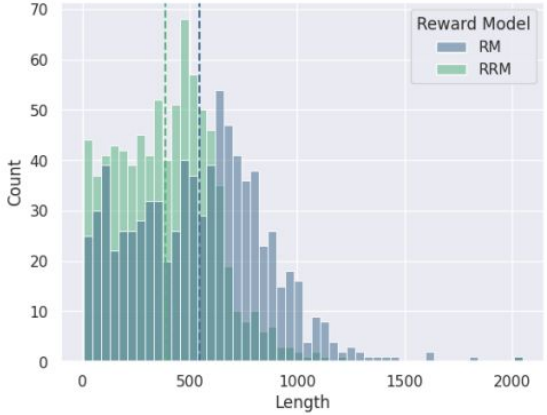
Figure 3: Distribution of response lengths on reward model training datasets. (a) the RM training data has longer chosen responses on average and not well calibrated (large percent deviation in left two bins between chosen and rejected) (b) the RRM training data is well calibrated and the average length of the chosen responses is even shorter than rejected. Additional neutral triplets can further calibrated the model. (c) Around 60% of chosen responses are longer in RM training data. On contrary, the lengths of chosen responses are more balanced in RRM training data.

Length Analysis - Policies

(a) Best of 8 responses



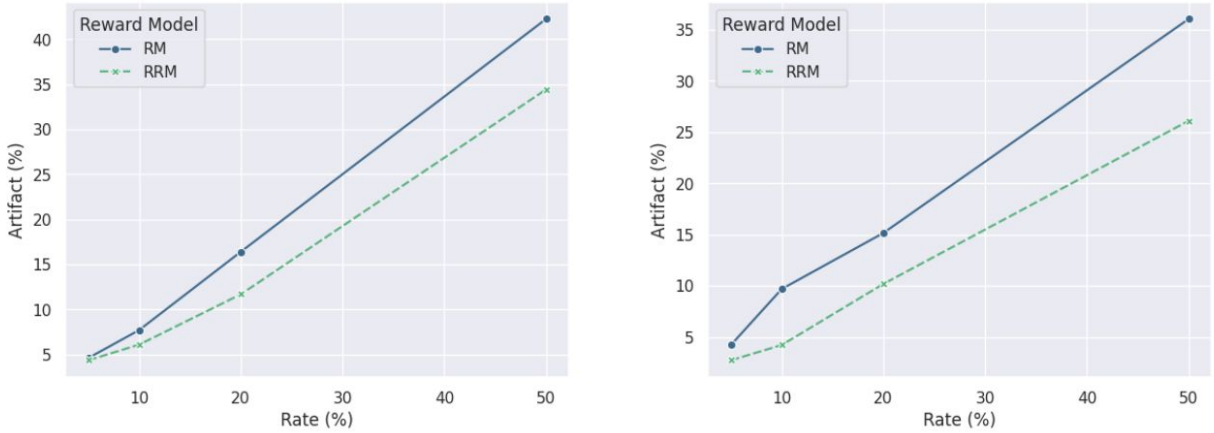
(b) Best of 64 responses



(c) DPO policy

Figure 4: Distribution of response lengths on AlpacaEval-2 prompts of various policies induced by RM and RRM, average length is marked by the dashed line. All policies show a lengthy bias towards longer responses for RM comparing with RRM.

Artifact Analysis



(a) Best of 8 responses

(b) Best of 64 responses

Figure 5: Proportion of BoN generated responses with artifact versus the rate of injected artifact. For each policy, we first sample N ($N = 8$ or 64) responses on AlpacaEval-2 prompts, then prepend "Sure, here is the response: " before each response with probability (Rate) 5%, 10%, 20%, 50%, respectively. Then we compute the proportion of BoN responses that have the above artifact (Artifact). The BoN policies induced by RRM are more robust to artifacts injected in the responses, suggesting that the proposed approach enables the model to focus more on the contextual signals instead of context-free artifacts in the reward model training data.

Google DeepMind

Thank you.

Google DeepMind