Real-Time Aligned Reward Model beyond Semantics

Zixuan Huang, Xin Xia, Yuxi Ren, Jianbin Zheng, Xuefeng Xiao, Hongyan Xie, Huaqiu Li,
Songshi Liang, Zhongxiang Dai, Fuzhen Zhuang, Jianxin Li, Yikun Ban, Deqing Wang

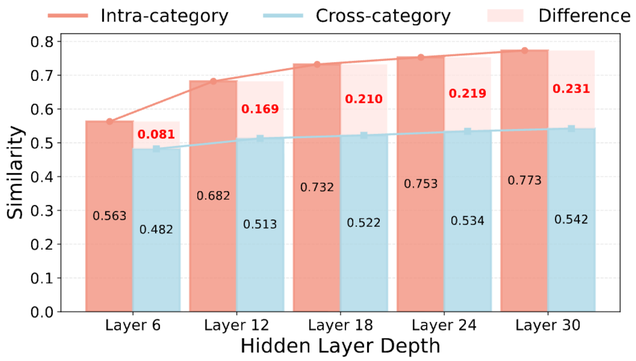
ICML
International Conference
On Machine Learning

MOTIVATION

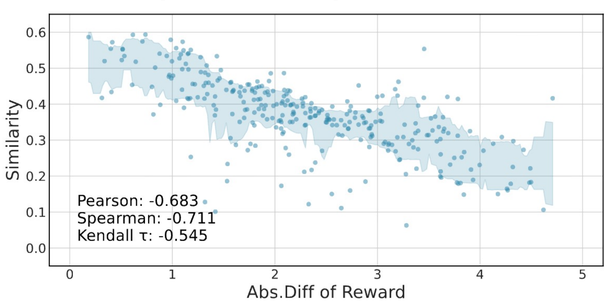
RLHF suffers from reward overoptimization because the policy can exploit spurious patterns in a static reward model as its distribution continuously shifts during training. To address this, we propose R2M, a lightweight RLHF framework that uses real-time policy feedback from evolving hidden states to align the reward model with the current policy distribution, thereby reducing reward discrepancy and improving reward reliability.

Can we design a new RLHF framework that preserves training efficiency while effectively achieving real-time alignment of reward models towards policy models?

OBSERVATIONS



Deep-layer hidden states effectively capture human preferences.



Higher hidden state similarity corresponds to smaller reward differences.

Theorem 3.1. *(Proof in Appendix A.1) Suppose that ϵ quantifies the extent of reward misalignment, we have the following upper bound of ϵ for R2M and vanilla RM:*

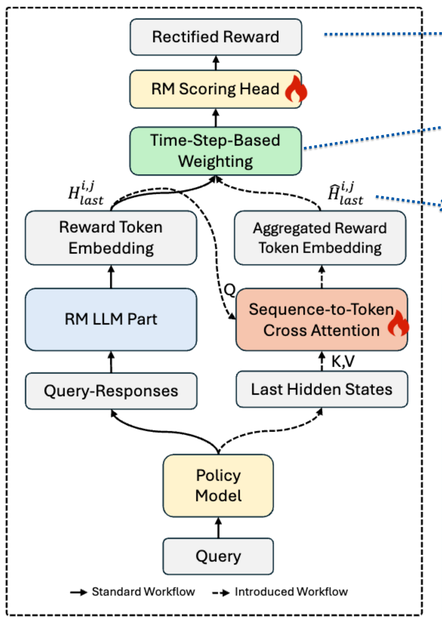
$$\epsilon_{R2M}^{(t)} \leq \left(1-\gamma^{(t)}\right)^{1/2} \cdot C + \Delta\mathcal{D}^{(t)} \cdot L$$

$$\epsilon_{vanilla}^{(t)} \leq C + \Delta\mathcal{D}^{(t)} \cdot L$$

where $\gamma^{(t)} \in [0,1]$, and $C > 0$.

METHODOLOGY

R2M Structure



$$r_\varphi(x_i, y_{i,j}, h_{i,j}) = \phi(H_{\text{fin}}^{i,j}) \in \mathbb{R}.$$

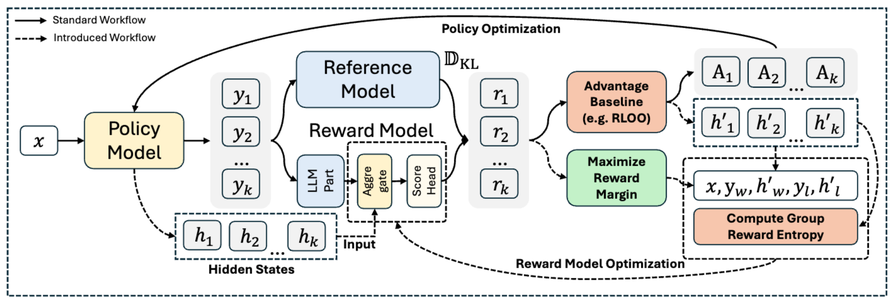
$$H_{\text{fin}}^{i,j} = (1-\omega(t))\hat{H}_{\text{last}}^{i,j} + \omega(t) H_{\text{last}}^{i,j},$$

$$\omega(t) = \max(\frac{1}{2}\cos(\frac{t}{T}\pi) + \frac{1}{2}, \Omega),$$

$$\hat{H}_{\text{last}}^{i,j} = \text{Softmax}\left(\frac{qk^T}{\sqrt{d}}\right) vW_o \in \mathbb{R}^{1\times D_{\text{rm}}},$$

The reward model is no longer a static scorer that relies solely on surface-level semantic representations; instead, it dynamically incorporates the policy model's internal states as real-time feedback, thereby better adapting to policy-induced distribution shifts and mitigating reward discrepancy and reward overoptimization.

Iterative Reward Model Lightweight Optimization



Overview of R2M. We first aggregate the last-layer hidden states hi from the policy with the LLM part output of the reward model. This aggregated representation is then fed into the scoring head for reward prediction. When the policy updates, we get the real-time feedback h'i and utilize it to construct preference pairs. Finally, we optimize the reward model by jointly minimizing the Bradley-Terry loss and the Group Reward Entropy loss.

$$\mathcal{L}_{\text{BT}}(i;\varphi) = -\log\sigma\big(r_\varphi(x_i, y_{i,w}, h_{i,w}) - r_\varphi(x_i, y_{i,l}, h_{i,l})\big),$$

$$\mathcal{L}_{\text{GRE}}(i;\varphi) = -\sum_{j=1}^{K} p_{i,j}\log p_{i,j}, \text{where}$$

$$p_{i,j} = \text{softmax}\left(\frac{r_{i,j} - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}\right),$$

$$\mathcal{L}_{\text{GREBT}}(i;\varphi) = (1-\alpha)\mathcal{L}_{\text{BT}}(i;\varphi) + \alpha\mathcal{L}_{\text{GRE}}(i;\varphi),$$

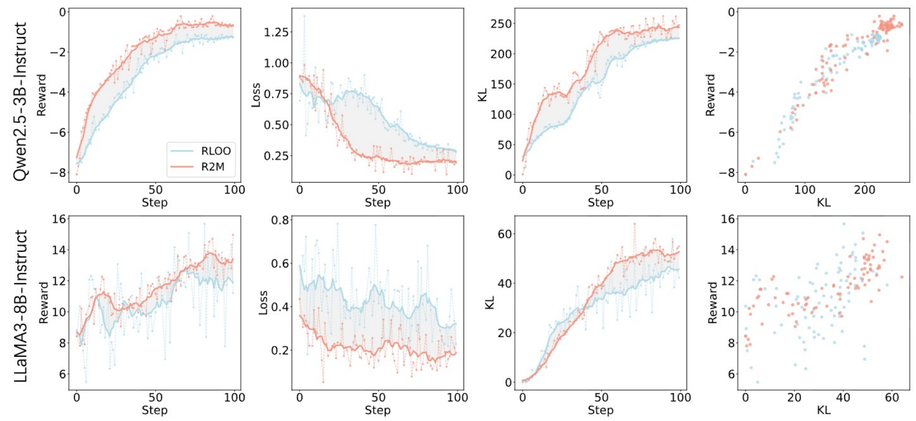
Theorem 4.1. *(Proof in Appendix A.2) Given $\varphi_\alpha = \arg\min_\varphi \mathcal{L}_{GREBT}(i;\varphi;\alpha)$ and the group degeneration degree $C_i(\varphi)$ for any group G_i, we establish the following results: (1) $C_i(\varphi_\alpha) < C_i(\varphi_0)$; (2) $\Delta C_i(\alpha) := C_i(\varphi_0) - C_i(\varphi_\alpha)$, $\forall\alpha_1 < \alpha_2$, $\Delta C_i(\alpha_1) < \Delta C_i(\alpha_2)$.*

EXPERIMENTS

Main Results on Dialogue Task

Method	Qwen2.5-3B-Instruct				LLaMA3-8B-Instruct			
	Alpaca-eval			MT-Bench	Alpaca-eval			MT-Bench
	LC(%)	WR(%)	LEN	GPT-4	LC(%)	WR(%)	LEN	GPT-4
SFT	15.5	15.8	2218	6.4	22.9	22.6	1899	6.9
ReMax	21.8 (↑ 40.6%)	25.1 (↑ 58.9%)	2916	6.4 (↑ 0.0%)	28.7 (↑ 25.3%)	30.7 (↑ 35.8%)	2289	7.0 (↑ 1.4%)
REINFORCE++	21.4 (↑ 38.1%)	26.4 (↑ 67.1%)	3252	6.3 (↓ 1.6%)	29.3 (↑ 27.9%)	31.8 (↑ 40.7%)	2192	6.8 (↓ 1.4%)
GRPO	22.7 (↑ 46.5%)	25.6 (↑ 62.0%)	3012	6.3 (↓ 1.6%)	29.5 (↑ 28.8%)	32.6 (↑ 44.2%)	2216	7.0 (↑ 1.4%)
+ R2M w/o Train	17.4 (↑ 12.3%)	19.4 (↑ 22.8%)	3317	6.2 (↓ 3.1%)	25.6 (↑ 11.8%)	27.0 (↑ 19.5%)	2261	6.7 (↓ 2.9%)
+ Pretrained RM	22.9 (↑ 47.7%)	28.2 (↑ 78.5%)	3101	6.4 (↑ 0.0%)	31.5 (↑ 37.5%)	34.3 (↑ 51.8%)	2278	7.1 (↑ 2.9%)
+ Iterative RM_{Head}	23.5 (↑ 51.6%)	27.8 (↑ 76.0%)	3050	6.5 (↑ 1.6%)	32.0 (↑ 39.7%)	33.9 (↑ 50.0%)	2250	7.0 (↑ 1.4%)
+ R2M	**25.8** (↑ 66.5%)	30.9 (↑ 95.6%)	2871	6.6 (↑ 3.1%)	**35.6** (↑ 55.4%)	**39.4** (↑ 74.3%)	2011	**7.3** (↑ 5.8%)
RLOO	21.9 (↑ 41.3%)	26.0 (↑ 64.6%)	3174	6.4 (↑ 0.0%)	28.4 (↑ 24.0%)	30.2 (↑ 33.6%)	2186	7.1 (↑ 2.9%)
+ R2M w/o Train	15.8 (↑ 1.9%)	20.5 (↑ 29.7%)	3154	6.2 (↓ 3.1%)	24.4 (↑ 6.5%)	27.4 (↑ 21.2%)	2366	6.5 (↓ 5.8%)
+ Pretrained RM	22.8 (↑ 47.1%)	27.4 (↑ 73.4%)	2992	6.5 (↑ 1.6%)	30.5 (↑ 33.2%)	32.2 (↑ 42.5%)	2172	7.1 (↑ 2.9%)
+ Iterative RM_{Head}	23.2 (↑ 50.3%)	27.0 (↑ 70.9%)	2950	6.5 (↑ 1.6%)	31.0 (↑ 35.4%)	31.8 (↑ 40.7%)	2150	7.0 (↑ 1.4%)
+ R2M	24.8 (↑ 60.0%)	**31.2** (↑ 97.5%)	2911	**6.7** (↑ 4.7%)	34.5 (↑ 50.7%)	38.2 (↑ 69.0%)	2011	**7.3** (↑ 5.8%)

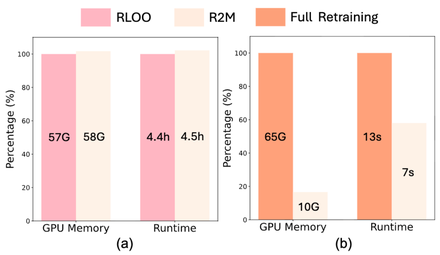
Training Dynamics



Ablation Study

Method	LC(%)	WR(%)	LEN
SFT	22.9	22.6	1899
RLOO	28.4 (↓ 17.7%)	30.2 (↓ 20.9%)	2186
+R2M w/ Noise	25.4 (↓ 26.4%)	26.4 (↓ 30.9%)	2276
+R2M w/o Train	24.4 (↓ 29.3%)	27.4 (↓ 28.3%)	2366
+R2M w/o BT	31.5 (↓ 8.7%)	35.7 (↓ 6.5%)	2116
+R2M w/o GRE	32.3 (↓ 6.4%)	36.2 (↓ 5.2%)	2191
+R2M	34.5	38.2	2011

Computational Cost



- ***R2M consistently improves RLOO/GRPO across dialogue and summarization tasks.***
- ***Ablations show that the gains arise from trained policy-feedback integration: frozen feedback degrades performance, while offline RM pretraining or iterative RM-head updates provide only limited improvements.***
- ***R2M produces more discriminative and robust reward signals, enabling larger effective policy updates while updating only lightweight modules rather than retraining the full RM.***