PIRA: Preference-Oriented Instruction-Tuned Reward Models with Dual Aggregation

Yongfu Xue
xueyongfu@outlook.com

Abstract

Reward models are pivotal for aligning Large Language Models (LLMs) with human preferences. Existing approaches face two key limitations: Discriminative reward models require large-scale annotated data, as they cannot exploit the preference instruction-following capability of LLMs available to generative reward models. Moreover, reward models are particularly prone to reward overoptimization, where LLMs exploit weaknesses in the reward function instead of improving true alignment. We introduce **PIRA**, a training paradigm that integrates three complementary strategies to address these challenges: (1) reformulating question–answer pairs into preference-task instructions to explicitly leverage LLMs' preference instruction-following capability, (2) averaging the rewards aggregated from diverse preference-task instructions for each sample, which mitigates task-specific bias and enhances robustness across evaluation perspectives, and (3) averaging outputs from the value head under different dropout rates to stabilize reward estimation. Experiments on public datasets show that PIRA improves performance considerably, enhances generalization, and effectively mitigates reward overoptimization.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) has become a standard approach for aligning large language models (LLMs) with human preferences (Ouyang et al., 2022; Bai et al., 2022). In this paradigm, human-labeled preference data is used to train a reward model, which then guides reinforcement learning to fine-tune the model. The accuracy and robustness of the reward model directly determine the effectiveness of RLHF (Ouyang et al., 2022; Touvron et al., 2023).

Despite its importance, training reliable reward models remains challenging. On the one hand, discriminative reward models often concatenate questions and answers directly without explicitly modeling the task intent (Dorka, 2024; von Werra et al., 2020; Liu et al., 2024; Havrilla et al., 2023). This underutilizes the instruction-following capabilities of LLMs, leading to inefficient data usage, as the model requires more data to implicitly learn latent task instructions that are not explicitly specified in the prompts. As a result, these models require large-scale annotation and frequently incur significant labeling costs. Generative reward models better leverage reasoning abilities and reduce annotation cost, but they introduce high inference latency due to autoregressive generation (Mahan et al., 2024; Zhang et al., 2024a). On the other hand, reward models also face the long-standing issue of reward overoptimization, where models exploit reward function flaws instead of achieving genuine alignment. Recent methods such as Thomas (Truong et al.) applies dropout at inference to model reward uncertainty, enabling broader exploration of user preferences and reducing the risk of local optima, at the cost of requiring a complete model inference for every dropout sample, and WARM (Ramé et al., 2024), which averages model weights to improve robustness under distribution shifts.

Based on these challenges, we introduce **PIRA**, which integrates three essential learning strategies: (1) reformulating question–answer pairs into preference-oriented task instructions, (2) averaging the rewards obtained from multiple reformulated preference-task instructions for each sample, which reduces instruction-specific bias and improves robustness, and (3) averaging the rewards generated under different dropout rates of the value head for each sample, which stabilizes reward estimation and further reduces variance. Our experiments on multiple public preference datasets show that PIRA performs better than traditional discriminative reward model methods. More importantly, when incorporated into the RLHF pipeline, PIRA-

Findings of the Association for Computational Linguistics: EACL 2026, pages 2226–2234
March 24-29, 2026 ©2026 Association for Computational Linguistics

trained reward models mitigate reward overoptimization, leading to better alignment with human preferences. Cross-domain evaluations further confirm that PIRA generalizes effectively across out-of-distribution tasks, highlighting its promise as a practical and scalable reward modeling paradigm.

2 PIRA: Methodology

Model decomposition. Let h_θ denote the backbone language model that maps a token sequence z to a representation $u = h_\theta(z)$, and let g_ψ denote a scalar value head that maps u to a reward. The overall reward model is $f_\phi = g_\psi \circ h_\theta$ with parameters $\phi = (\theta, \psi)$.

Preference Instruction Set The Preference Instruction Set $\mathcal{T} = t_1, \ldots, t_{K_{\text{all}}}$, where $K_{\text{all}} = |\mathcal{T}|$, consists of evaluation instructions that are generated by a large language model and subsequently refined through human review (see Appendix F). Each instruction serves as a holistic rubric for assessing model responses, rather than a list of dimension-specific criteria. Variations in phrasing and emphasis across instructions provide complementary perspectives, leading to a more balanced evaluation. Functionally, $\mathcal{T}$ acts as a generative evaluation prompt for a discriminative reward model. In generative architectures like GPT, this final state integrates prompt and content information, yielding an embedding that effectively supports downstream discriminative or evaluative tasks.

Preference-Oriented Instruction Reformulation Given a prompt x with two possible responses, we denote the preferred response as y^c and the rejected one as y^r. To make the reward model explicitly aware of the preference task, we prepend a task-specific instruction $t \in \mathcal{T}$ to the input, forming

$$r_\phi(x, y \mid t) = f_\phi([t; x; y]) = g_\psi\big(h_\theta([t; x; y])\big).$$

Since the reward head is randomly initialized, it initially lacks any understanding of scoring and must learn preferences entirely from data. The input formulation thus becomes crucial for stable learning. Conventional approaches simply concatenate the question and answer without clarifying that the task involves scoring. As a result, the model must simultaneously learn both the concept of scoring and how to apply it, which increases training difficulty.

Training Objective We train the model using the standard Bradley–Terry objective (Bradley and Terry, 1952). For a preference dataset $\mathcal{D} = \{(x_i, y_i^c, y_i^r)\}_{i=1}^N$, the loss function is

$$\mathcal{L}(\phi) = -\sum_{(x_i, y_i^c, y_i^r) \in \mathcal{D}} \log \sigma(r_\phi(x_i, y_i^c \mid t) - r_\phi(x_i, y_i^r \mid t))$$

where $\sigma(\cdot)$ is the logistic function, and t is randomly sampled from $\mathcal{T}$ for each instance. Dropout is applied to both g_ψ and h_θ.

The backbone h_θ and reward head g_ψ are updated with different learning rates to reflect their distinct roles. Since h_θ already encodes rich linguistic knowledge from pretraining, it is fine-tuned conservatively to maintain stability and avoid catastrophic forgetting. Conversely, g_ψ, a lightweight module, uses a higher learning rate for faster adaptation to preference signals and more effective supervision.

Inference-Time Instruction-Set Reward Averaging Rather than using a single preference instruction to evaluate each sample, PIRA computes rewards by averaging over multiple instructions $\mathcal{T}_K = \{t_1, \ldots, t_K\}$, $\mathcal{T}_K \subseteq \mathcal{T}_{(all)}$. For a given (x, y), the aggregated reward is

$$R_{\text{inst}}(x, y) = \frac{1}{K} \sum_{k=1}^{K} r_\phi(x, y \mid t_k),$$

where $r_\phi(x, y \mid t_k)$ is the reward predicted under instruction t_k. This averaging incorporates diverse evaluative perspectives, reducing prompt-specific bias and enhancing robustness.

Inference-Time Stochastic Value-Head Averaging PIRA improves estimation stability by applying Monte Carlo dropout (Gal and Ghahramani, 2016) only to the value head g_ψ. Unlike standard Monte Carlo dropout with a fixed rate, PIRA varies the dropout rate to strengthen robustness and generalization. For a preference instruction t,

$$r^{(m)}(x, y \mid t) = g_\psi^{(\delta_m)}\big(h_\theta([t; x; y])\big), \quad m = 1, \ldots, M.$$

δ_m denotes the m-th dropout rate within g_ψ. The final reward is the mean of these samples:

$$R_{\text{stoc}}(x, y \mid t) = \frac{1}{M} \sum_{m=1}^{M} r^{(m)}(x, y \mid t).$$

Model	Method	HH	Oasst	SHP	UltraFeedback	Alpaca-farm	HH-cleaned	Average
Mistral-7B-v0.1	Baseline	63.3	72.6	66.9	65.5	57.9	68.1	65.7
	Thomas	63.2	72.0	67.1	66.4	56.7	67.9	65.6
	Thomas*	63.8	**73.6**	69.1	72.0	58.8	77.5	69.1
	WARM	63.7	72.6	67.2	67.1	58.1	68.2	66.2
	WARM*	**64.2**	73.3	69.8	**72.4**	59.3	79.9	69.8
	PIRA	64.0	73.3	**70.9**	71.3	**60.4**	**80.4**	**70.1**
LLaMA3-8B	Baseline	61.4	72.0	67.2	67.5	57.6	64.3	65.0
	Thomas	61.7	72.3	68.1	67.0	56.8	64.9	65.1
	Thomas*	64.9	73.5	69.4	**70.5**	59.9	**76.6**	69.1
	WARM	62.4	72.5	68.5	67.8	57.2	64.7	65.5
	WARM*	66.3	74.8	70.2	68.5	60.7	75.2	69.3
	PIRA	**66.8**	**75.5**	**70.5**	69.3	**61.2**	75.5	**69.8**
Qwen2.5-1.5B	Baseline	58.6	68.5	64.6	62.6	55.6	62.6	62.1
	Thomas	59.6	67.2	64.2	61.8	56.5	62.6	62.0
	Thomas*	64.3	70.6	66.4	**69.6**	59.7	63.0	65.6
	WARM	59.2	68.2	64.9	62.2	56.9	62.4	62.3
	WARM*	65.3	70.3	**67.3**	69.4	**60.2**	**64.9**	**66.2**
	PIRA	**66.0**	**71.5**	67.0	68.0	59.6	64.6	66.1
Qwen2.5-7B	Baseline	60.2	69.2	66.0	64.6	57.4	63.4	63.5
	Thomas	60.4	71.3	65.5	64.9	56.9	65.0	64.0
	Thomas*	64.4	71.9	67.2	70.6	60.6	76.9	68.6
	WARM	61.0	71.0	66.3	65.3	58.1	66.2	64.7
	WARM*	**66.9**	72.7	67.7	71.7	61.3	76.7	69.5
	PIRA	66.4	**73.0**	**68.0**	**72.0**	**62.6**	**77.0**	**69.8**

Table 1: Performance comparison across various models and datasets.

During inference, h_θ performs a single deterministic forward pass (without dropout), while multiple stochastic passes through g_ψ yield an ensemble estimate at minimal computational cost.

Dual aggregation The final reward score is computed as:

$$\begin{aligned} R(x,y) &= \frac{1}{K}\sum_{k=1}^{K} R_{\text{stoc}}(x,y \mid t_k) \\ &= \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{M}\sum_{m=1}^{M} r^{(m)}(x,y \mid t_k)\right). \end{aligned}$$

We sample $\delta_m \sim \text{Uniform}(0.1, 0.4)$ and choose small K and M (e.g., $K \leq 6$, $M \leq 12$) to balance stability and efficiency. This two-level averaging—across preference instructions and stochastic value-head realizations—reduces estimator standard deviation, enhances robustness, and mitigates reward overoptimization.

3 Experiments

3.1 Experimental Setup

We conduct experiments with LLaMA3-8B (Dubey et al., 2024), Mistral-7B-v0.1 (Jiang et al., 2023), and Qwen2.5 (1.5B, 7B) (Qwen et al., 2025). Training and evaluation use multiple preference datasets, including HH (Bai et al., 2022), HH-cleaned (Wang et al., 2024a), SHP (Ethayarajh et al., 2022), Alpaca-farm (Dubois et al., 2023), Oasst (Köpf et al., 2023), and UltraFeedback (Cui et al., 2023).

We fine-tune using LoRA (Hu et al., 2022) (rank 128, $\alpha = 128$, dropout 0.05), added to the query and value projection layers. We also optimize the value head parameters. Optimization uses AdamW with batch size 32, 2 epochs, and a 0.05 warmup ratio. Learning rates are 1×10^{-6} for h_θ and 5×10^{-4} for g_ψ. We set $K = 6$ and $M = 12$.

All experiments run on NVIDIA A800 GPUs, with results averaged over three seeds (42, 22, 33).

3.2 Main Results

Performance Comparison We evaluate PIRA against the baseline and Thomas (Truong et al.) methods. The baseline employs simple concatenation of questions and answers, and Thomas method performs sampling using different dropout masks (dropout rate = 0.25) with 4 forward passes, as described in Thomas's paper.

In addition, we conducted further experiments comparing PIRA with WARM (Ramé et al., 2024) following the methodology proposed in the original paper. Checkpoints were saved after the 1st, 2nd,

Method	K	M	HH-cleaned Acc. ↑	SHP Acc. ↑	Standard Deviation ↓
Baseline	–	–	64.3	67.2	2.1
+ Preference-oriented Task Instruction	–	–	73.0	69.2	1.6
+ Instruction-Set Averaging	4	–	73.9	69.5	1.1
+ Value-Head Stochastic Averaging	4	4	75.1	70.2	0.8
PIRA (both)	**6**	**12**	**75.5**	**70.5**	**0.7**

Table 2: The effects of different parts of PIRA on accuracy and stability.

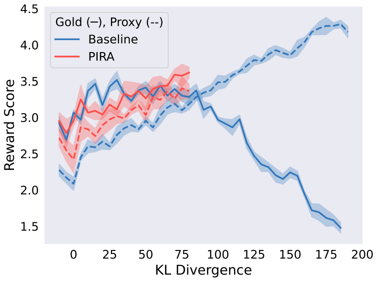

Figure 1: Baseline vs. PIRA

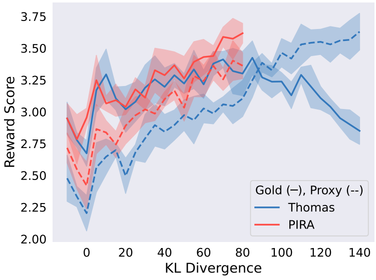

Figure 2: Thomas* vs. PIRA

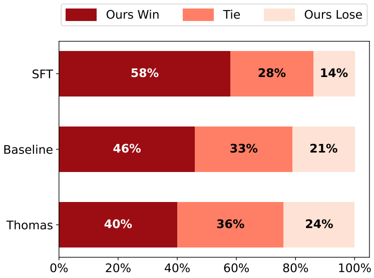

Figure 3: End-to-End Evaluation

and 3rd epochs, and the final reward model was obtained by averaging the parameters of these three checkpoints.

As shown in Table 1, * denotes the corresponding method augmented with preference-oriented instruction reformulation, as used in PIRA. The results in the table indicate that PIRA generally outperforms the baseline Thomas and WARM methods.

Reward Hacking Mitigation During PPO training, both the reward and policy models were trained using Qwen2.5-1.5B, while the gold reward model used Qwen2.5-7B. Training employed the Alpacafarm dataset and used a small KL penalty (0.0005). For more experimental details and additional results, please refer to Appendix D. In baseline PPO runs, we observed sharp spikes in KL divergence and rapid reward inflation (Figure 1 and Figure 2). The gold rewards initially increase but later decline, indicating potential reward hacking. By contrast, PIRA-trained models keep the policy close to the data-supported region: KL divergence remains bounded, reward inflation is suppressed, and gold rewards improve monotonically over training rather than peaking early and collapsing.

3.3 Ablation Studies

Joint Effect of Instruction and Stochastic Averaging Using LLaMA3-8B, Table 2 shows that combining instruction-set averaging and value-head stochastic averaging (PIRA; (K=6, M=12)) achieves the best overall results—highest HH-cleaned (75.5) and SHP (70.5) accuracies, and lowest standard deviation (0.7) for HH-cleaned. The preference-oriented task instruction contributes most significantly to PIRA, while instruction-set averaging and value-head stochastic averaging more effectively reduce the standard deviation in predicted rewards.

Impact of Instruction-Set Averaging and Value-Head Stochastic Averaging We examine how varying the number of preference instructions (K) and stochastic forward passes (M) affects LLaMA3-8B's performance. As shown in Table 3, increasing K from 1 to 6 significantly reduces accuracy standard deviation and generally improves results. Similarly, Table 4 demonstrates that larger M values enhance both stability and accuracy. Because dropout is applied only to the lightweight value head, the computational overhead remains minimal, resulting in an approximately 7% increase in latency when $M = 12$. When $K = 6$, however, the computational cost increases sixfold. This additional cost must be weighed against the corresponding benefits, including reduced bias and improved robustness.

K	HH-cleaned Acc. ↑	Standard Deviation ↓
1	73.2	1.8
2	73.6	1.6
4	73.5	1.5
6	74.4	1.2

Table 3: Impact of Instruction-Set Averaging.

M	HH-cleaned Acc. ↑	Standard Deviation ↓	Latency ↑ (%)
1	72.9	1.9	+0
2	72.9	1.8	+2
4	73.7	1.3	+3
8	74.0	1.3	+5
12	74.2	1.0	+7

Table 4: Impact of Value-Head Stochastic Averaging.

End-to-End Evaluation We perform pairwise human-like preference evaluations using GPT-4o as the evaluator to assess model alignment quality after RLHF. Each comparison between the PIRA-optimized policy and the reference models (SFT, Baseline, and Thomas*) is repeated three times under randomized prompts. Final win rates are computed via majority voting across evaluation rounds. Results indicate that the PIRA-optimized policy consistently outperforms all baselines (Figure 3).

Data Efficiency and Generalization Under varying data scale settings, PIRA demonstrates strong adaptability: it yields notable benefits in low-data scenarios and maintains stable gains as data increases (see Appendix A). When evaluated on cross-dataset generalization, PIRA shows robust performance under distribution shifts, with response length emerging as a key influencing factor (see Appendix B). Furthermore, scaling experiments on the Llama-2-13B model show that PIRA transfers effectively to larger models, achieving competitive performance compared to other methods (see Appendix C).

4 Conclusion

PIRA introduces a simple yet effective framework for building robust reward models. By combining instruction reformulation and dual aggregation, it enhances stability, reduces bias, and mitigates reward overoptimization. Experiments show consistent gains across models and datasets, making PIRA a practical solution for preference-aligned LLM training.

Limitations

PIRA has not yet been evaluated on larger-scale language models, and its scalability to models beyond 13B parameters remains to be investigated. Additionally, value-head stochastic averaging introduces a slight inference overhead, while instruction-set averaging incurs a more substantial overhead.

References

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*.

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*, abs/2204.05862.

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.

Lu Chen, Rui Zheng, Binghai Wang, Senjie Jin, Caishuang Huang, Junjie Ye, Zhihao Zhang, Yuhao Zhou, Zhiheng Xi, Tao Gui, and 1 others. 2024. Improving discriminative capability of reward models in rlhf using contrastive learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15270–15283.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2023. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, and 1 others. 2023. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*.

Nicolai Dorka. 2024. Quantile regression for distributional reward models in rlhf. *ArXiv preprint*, abs/2409.10164.

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

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, and 1 others. 2023. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings* *of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Alexander Havrilla, Maksym Zhuravinskyi, Duy Phung, Aman Tiwari, Jonathan Tow, Stella Biderman, Quentin Anthony, and Louis Castricato. 2023. trlx: A framework for large scale reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8578–8595.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, and 1 others. 2023. Openassistant conversations-democratizing large language model alignment. *Advances in neural information processing systems*, 36:47669–47681.

Shujun Liu, Xiaoyu Shen, Yuhang Lai, Siyuan Wang, Shengbin Yue, Zengfeng Huang, Xuanjing Huang, and Zhongyu Wei. 2024. Haf-rm: A hybrid alignment framework for reward model training. *ArXiv preprint*, abs/2407.04185.

Xingzhou Lou, Dong Yan, Wei Shen, Yuzi Yan, Jian Xie, and Junge Zhang. 2024. Uncertainty-aware reward model: Teaching reward models to know what is unknown. *arXiv preprint arXiv:2410.00847*.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. 2024. Generative reward models. *ArXiv preprint*, abs/2410.12832.

Ted Moskovitz, Aaditya K Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca D Dragan, and Stephen McAleer. 2023. Confronting reward model overoptimization with constrained rlhf. *arXiv preprint arXiv:2310.04373*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. 2024. Warm: On the benefits of weight averaged reward models. *arXiv preprint arXiv:2401.12187*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Sang T Truong, Duc Quang Nguyen, Tho Quan, and Sanmi Koyejo. Thomas: Learning to explore human preference via probabilistic reward model. In *ICML 2023 Workshop The Many Facets of Preference-Based Learning*.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, and 1 others. 2024a. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024b. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10582–10592, Miami, Florida, USA. Association for Computational Linguistics.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024. Regularizing hidden states enables learning generalizable reward model for llms. *Advances in Neural Information Processing Systems*, 37:62279–62309.

Ziyi Ye, Xiangsheng Li, Qiuchi Li, Qingyao Ai, Yujia Zhou, Wei Shen, Dong Yan, and Yiqun Liu.

2024. Beyond scalar reward model: Learning generative judge from preference data. *ArXiv preprint*, abs/2410.03742.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024a. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*.

Xiaoying Zhang, Jean-Francois Ton, Wei Shen, Hongning Wang, and Yang Liu. 2024b. Overcoming reward overoptimization via adversarial policy optimization with lightweight uncertainty estimation. *arXiv preprint arXiv:2403.05171*.

A Impact of Data Scale on Model Performance

Figure 4 shows that PIRA yields the largest relative gains in low-data regimes (100–1000 examples). For example, at 500 HH examples, PIRA improves accuracy by +9 over the baseline. With larger datasets ($\geq$ 10k), performance converges but PIRA maintains a consistent margin.

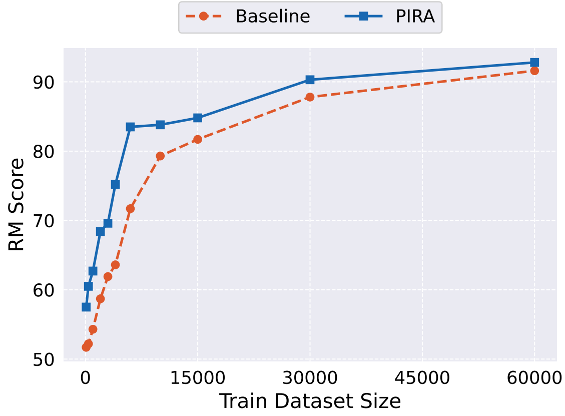

Figure 4: Impact of training data size on model performance: baseline vs. PIRA.

B Cross-Dataset Generalization

We evaluate on OOD test sets to assess robustness. As shown in Figure 5, PIRA consistently outperforms the baseline on Oasst and HH-cleaned, confirming adaptability under distribution shift. Performance degrades on UltraFeedback, whose longer and more complex responses reduce instruction prompt effectiveness. Length-controlled subsets confirm that the drop is largely due to response length, highlighting an area for future work.

C Performance on Larger Models

Based on Table 5, PIRA consistently improves over the baseline on the larger Llama-2-13B model. It maintains a clear advantage across benchmarks, demonstrating stable performance and good scalability to larger models.

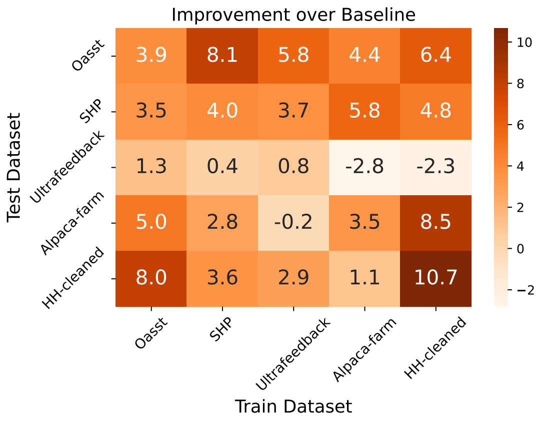

Figure 5: Performance improvements achieved by the PIRA method over the baseline across cross-dataset combinations.

D PPO Training Configurations

The PPO training configuration is provided in Table 6.

E Related work

Large language models (LLMs) often produce harmful, biased, or inappropriate content. A primary method for mitigating these issues is reinforcement learning from human feedback (RLHF). RLHF employs reward models to approximate human preferences and guide model behavior toward desirable outcomes. These reward models can be discriminative, generative, or hybrid in nature. Discriminative reward models, based on the Bradley–Terry framework (Bradley and Terry, 1952), use a value head to generate scalar rewards, whereas generative reward models leverage LLM output probabilities or voting mechanisms. As these reward models scale, they face the challenge of reward overoptimization, where model signals diverge from genuine human intent, making robustness a central concern in alignment research.

E.1 Reward Models

Discriminative reward models have given rise to many expressive architectures. ArmoRM (Wang et al., 2024b) improves interpretability with multi-objective regression, HAF-RM (Liu et al., 2024) jointly trains reward and generation heads, and QRM (Dorka, 2024) models full reward distribu-

Method	HH	Oasst	SHP	UltraFeedback	Alpaca-farm	HH-cleaned	Average
baseline	63.8	73.6	68.7	68.5	59.6	67.1	66.9
PIRA	70.2	76.8	72.3	72.8	63.4	75.3	71.8

Table 5: Results on the Llama-2-13B Model

Rollout samples	256
Chunk size	64
Sampling temperature	1.0
PPO epochs	4
Batch size	16
Total training steps	2500
Clip range (ϵ)	0.2
KL penalty coefficient	0.0005
lambda for GAE	0.95
Optimizer	AdamW
Learning rate	2×10^{-5}
Adam β_1 / β_2	0.95 / 0.99
Weight decay	1×10^{-6}
LR scheduler	Cosine annealing
Random seed	42

Table 6: PPO training configuration.

tions to capture preference diversity. Contrastive learning further strengthens reward representations (Chen et al., 2024). Overall, these methods typically concatenate the question and answer pairs from preference data as joint inputs to the reward model. Generative reward models, though less studied, leverage LLMs' generative capacity. GenRM integrates chain-of-thought reasoning (Mahan et al., 2024), the Generative Validator reframes validation as prediction (Zhang et al., 2024a), and Con-J trains generative judges to provide explanatory, robust judgments (Ye et al., 2024). These methods introduce more inference latency.

Some models combine discriminative and generative methods. The CLoud (Ankner et al., 2024) generates a critique of response quality and then predicts a scalar reward using the input, response, and critique. By leveraging language generation, it makes response quality explicit, similar to a chain of thought, addressing the limitations of traditional implicit reward models.

E.2 Reward Overoptimization

Reward overoptimization occurs when the reinforcement learning agent exploits flaws in the reward model (RM), achieving high proxy scores while degrading true alignment with human intent (Gao et al., 2023). To mitigate this effect, prior work has taken two main directions. The first focuses on optimization constraints that keep policy updates close to the preference data manifold (Moskovitz et al., 2023). These approaches limit reward hacking by restricting the divergence between optimized policies and human-preferred behaviors, yet often underexploit the model's full learning potential. The second line of work aims to improve RM robustness. Ensemble methods provide partial robustness by aggregating multiple RMs (Eisenstein et al., 2023; Coste et al., 2023), while weight averaging smooths reward landscapes and enhances stability under distribution shifts (Ramé et al., 2024). Recent advances further incorporate uncertainty estimation to regularize or guide policy learning, such as uncertainty-aware RMs (Lou et al., 2024), hidden-state regularization (Yang et al., 2024), and adversarial optimization under uncertainty (Zhang et al., 2024b). However, these methods are often complex and difficult to apply in practice. In contrast, we propose a simple yet effective approach that mitigates reward overoptimization while maintaining robust performance.

F Instructions

The preference task instruction consists of three parts: the preference task instruction description, represented by t; the question slot, represented by q; and the answer slot, represented by a.

Here is a complete preference instruction template:

Preference Task Instruction Template

$\{t\}$

question:
$\{q\}$

response:
$\{a\}$

The following list enumerates all t values used in the experiments:

1. Evaluate whether the response demonstrates a reliable grasp of facts and reasoning. Consider if any parts appear speculative, misleading, or inconsistent with known information. Provide your overall judgment of its truthfulness.

2. Judge the extent to which the response stays aligned with the user's intent and avoids digressions. Highlight any parts that seem tangential or dilute the main purpose of the answer.

3. Determine whether the response clearly fulfills the user's underlying intent. Focus on how directly and understandably it addresses the core question without overcomplicating or oversimplifying.

4. Assess whether the response demonstrates logical flow, internal consistency, and coherent argumentation. Note if reasoning steps are missing, circular, or contradictory.

5. Evaluate the response's utility for a general user. Does it offer sufficient detail, context, and actionable insight? Consider whether it is understandable to non-experts and avoids unnecessary technical jargon.

6. Examine how well the response justifies its claims. Does it provide reasoning that is sound, evidence-based, and transparent? Identify whether the explanation shows deep understanding or shallow paraphrasing.

7. Assess the clarity, conciseness, and stylistic balance of the response. Is it fluent and engaging while remaining precise? Note both stylistic strengths and weaknesses that affect readability.

8. Reflect on how faithfully the response follows the original instruction while maintaining creative, well-structured expression. Evaluate whether the tone, structure, and style enhance or detract from the intent.

9. Analyze the response critically: what are its major strengths, weaknesses, and possible risks if used in a real context (e.g., misunderstanding, misinformation, or harm)? Offer a short paragraph on each.

10. Provide an overall evaluation of the response's completeness, tone, and coherence. Discuss whether it could be improved, and if so, which aspects—content, logic, or presentation—should be prioritized.