

Rethinking the Role of Proxy Rewards in Language Model Alignment

Anonymous ACL submission

Abstract

Learning from human feedback via proxy reward modeling has been studied to align Large Language Models (LLMs) with human values. However, achieving reliable training through that proxy reward model (RM) is not a trivial problem, and its behavior remained as a black-box. In this paper, we study the role of proxy rewards in the LLM alignment via ‘reverse reward engineering’ by composing interpretable features as a white-box reward function. We aim to replicate the ground truth (gold) reward signal by achieving a *monotonic* relationship between the proxy and gold reward signals after training the model using the proxy reward in reinforcement learning (RL). Our findings indicate that successfully emulating the gold reward requires generating responses that are relevant with enough length to open-ended questions, while also ensuring response consistency in closed-ended questions. Furthermore, resulting models optimizing our devised white-box reward show competitive performances with strong open-source RMs in alignment benchmarks. We highlight its potential usage as a simple but strong reward baseline for the LLM alignment, not requiring explicit human feedback dataset and RM training.

1 Introduction

To align large language models (LLM) with human values like helpfulness, human feedback-based learning has been studied (Ouyang et al., 2022; Bai et al., 2022a,b; Rafailov et al., 2023). Reinforcement Learning from Human Feedback (RLHF) is a dominant approach to exploiting human feedback. Typically, the human feedback is used to train a proxy reward model (RM), and a policy model is optimized over the reward signal from the RM using RL (Schulman et al., 2017; Ziegler et al., 2020).

However, reliable RL training with the proxy RM is not a trivial problem. Gao et al. (2023) study the overoptimization problem of the policy training against the imperfect RM. It is a phenomenon

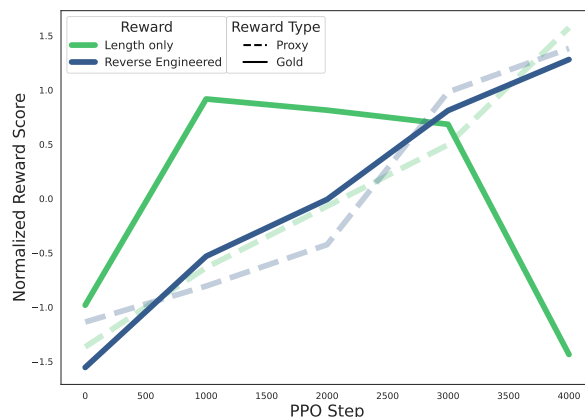


Figure 1: A preview of our reverse reward engineering experiment. First, we design reward functions with interpretable features such as the length or relevance of the response. Then, we conduct RL training using each of the designed functions as a proxy reward and deem it a success in reverse engineering if a *monotonic* relationship between the proxy reward and the ground truth (Gold) reward is observed in the validation set. In this plot, the reverse-engineered reward (blue) exhibits such a tendency, whereas the length-only reward (green) does not achieve the monotonic relationship.

that the proxy rewards from the RM are increased continuously, but the true rewards are saturated or even decreased in fact. Similarly, the policy model often finds undesired shortcuts in the reward signal from the imperfect black-box RM, i.e., “reward hacking” (Skalse et al., 2022; Pang et al., 2022).

A line of research has studied the innate limitations of human feedback. Xu et al. (2023) reveal prior knowledge affects the preference judgment in that crowd workers often choose the preferred response considering surface-level properties such as conciseness and specificity, while experts focus on more essential properties like factuality and completeness. Similarly, Hosking et al. (2023) further categorize the attributions of the preference judgments and confirm that human judgments could be biased by stylistic features like assertiveness. Moreover, Singhal et al. (2023) analyze the correlations

between long response and RLHF, demonstrating a verbosity bias in the human feedback.

In this work, we show merely combining interpretable features as a proxy reward function can substantially maximize the score of state-of-the-art RM (Lambert et al., 2024) and further enhance human preferences in various alignment benchmarks (Chiang et al., 2023; Li et al., 2023; Zheng et al., 2023). Specifically, we study the role of proxy rewards learned by human feedback in the RLHF by employing “reverse reward engineering” with interpretable features, such as length or relevance of the responses, to compose the white-box reward function, as shown in Figure 1. We define the goal of reverse engineering as achieving a *monotonic* relationship between the proxy and ground truth (gold) reward signals after training the model with the proxy reward in an RL manner.

Experimental results indicate that the key to imitating the gold reward signal lies in generating responses that are sufficiently lengthy yet relevant and faithful for open-ended queries while ensuring consistency in responses to closed-ended queries. Contrary to previous study (Singhal et al., 2023), our results demonstrate that solely optimizing towards lengthy response makes drastic drops in the gold reward, showing severe overoptimization. Also, the reward branching according to query type reliably improves the win rate while not introducing unnecessary verbosity compared to other baseline reward options. Furthermore, we find that the reward differentiation brings advantages in improving preference while minimizing alignment tax (Ouyang et al., 2022), a phenomenon where the increased preference accompanies the degraded performance on other NLP tasks. Notably, the engineered white-box reward often results in even more improvements than the strong open-source RM like UltraRM-13B (Cui et al., 2023) and also generally works well across LLM backbones, demonstrating its potential usage as a baseline reward.

Our contributions are summarized in three folds:

- We investigate the role of proxy rewards learned by human feedback via reverse reward engineering with interpretable features.
- Our results suggest that the key to replicating the gold reward involves producing responses that are not only lengthy but also relevant for open-ended queries while maintaining consistency in responses to closed-ended queries.
- We highlight the potential usage of the reverse-

engineered white-box reward function as a simple but strong reward baseline not requiring explicit human feedback dataset.

2 Related Work

Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback (RLHF) is the most prevalent approach to leverage the human preference (Ziegler et al., 2020; Ouyang et al., 2022; Bai et al., 2022a; Liu et al., 2022; Scheurer et al., 2023). These approaches include the reward modeling stage to develop a proxy RM predicting the human preference over the response, e.g., typically as a scalar format. Then, the trained proxy RM computes reward scores for the sampled responses in the RL stage, i.e., rollout. Recently, replacing human feedback with AI or synthetic feedback has been studied (Bai et al., 2022b; Kim et al., 2023; Sun et al., 2023b; Cui et al., 2023). There are attempts to leverage multiple rewards beyond the overall reward (Touvron et al., 2023; Wu et al., 2023; Rame et al., 2023; Jang et al., 2023). Wu et al. (2023) devise fine-grained rewards consisting of relevance, factuality, and information completeness to enhance long-form question-answering.

Pitfalls of Reward Modeling and Human Feedback

RLHF faces various challenges, especially in reward modeling and human feedback itself (Casper et al., 2023). Gao et al. (2023) present the overoptimization problem by the imperfect proxy reward models. The issue arises when the policy, tailored to enhance the estimated rewards, fails to improve real-world rewards. Many studies try to mitigate the overoptimization problem (Moskowitz et al., 2023; Coste et al., 2023). Similarly, Shen et al. (2023) study reward inconsistency and leverage a sentence encoder to increase the reward consistency. The work leverages SimCSE (Gao et al., 2021) to augment the preference dataset and normalize reward scores. The policy models within the RLHF often find and exploit undesired shortcuts or errors in the RMs to maximize the proxy rewards. The phenomenon is known as reward hacking (Amodei et al., 2016; Pan et al., 2022; Pang et al., 2022; Song et al., 2023; Eisenstein et al., 2023). Much literature points out that human preference can be easily biased by superficial features such as complexity and assertiveness (Xu et al., 2023; Hosking et al., 2023). Singhal et al. (2023) study a high correlation between response length and RLHF training. Motivated by

these studies, we integrate a set of interpretable features, such as length incentive, repetition penalty, query relevance, and distinction by query type, to understand the behaviors of the learned proxy reward models.

3 Experimental Setup

3.1 Reverse Reward Engineering

Inspired by previous studies on superficial biases in human feedback (Singhal et al., 2023; Xu et al., 2023; Hosking et al., 2023), we aim to explore whether there would be a reward function with interpretable features that can maximize the “ground truth” reward signal in reverse. Started by the length bias (Singhal et al., 2023; Chen et al., 2024), we progressively design the additional features to achieve the goal. More formally, we first perform RL training, i.e., Proximal Policy Optimization (PPO) (Schulman et al., 2017), with the designed function as a proxy reward and then consider it a success in reverse engineering if a *monotonic* relationship between the proxy reward and the ground truth reward is observed in the validation set. In other words, if overoptimization (Gao et al., 2023) for the proxy reward is detected, it will be considered a failure in reverse engineering.

3.2 Features for White-box Reward

In this subsection, we detail the reward features employed in our experiments on reverse engineering. For a set of (x, y) pairs, we denote x as a query, y as a reference response, and \hat{y} as a rollout response of Proximal Policy Optimization (PPO) (Schulman et al., 2017; Ziegler et al., 2020).

Length Incentive (LI). First, we introduce Length Incentive (LI) to exploit the verbosity bias in human preference (Singhal et al., 2023; Wang et al., 2024; Chen et al., 2024). It scores the reward by calculating $LI(\hat{y}) = \frac{\# \text{ words of } \hat{y}}{100}$. Intuitively, this reward function promotes the longer generation regardless of the given query x . It is similar to LPPO suggested by (Singhal et al., 2023), but we do not normalize the reward with the target length.

Repetition Penalty (RP). In our preliminary study, solely using LI as a reward function produces undesired repetitions in the generations. We employ the Repetition penalty (RP) to mitigate unnecessary repetitions in \hat{y} . It checks the unique trigram ratio of \hat{y} , i.e., $RP(\hat{y}) = \frac{\# \text{ unique trigram of } \hat{y}}{\# \text{ trigrams of } \hat{y}}$. Integrating it with the Length Incentive function effi-

ciently prevents redundancy and encourages proper length of outputs, i.e., $LI(\hat{y}) \cdot RP(\hat{y})$.

Query Relevance (QR). We find that combining LI and RP is somewhat effective, but the functions often promote the generation of irrelevant content since both functions are input-agnostic. Hence, we involve the Query Relevance (QR) as one of the reward functions. It checks whether the generated response \hat{y} contains off-topic contents for a provided query x . We compute the relevance score between the query and sampled response: $QR(x, \hat{y}) = M(x) \cdot M(\hat{y}) \in \mathbb{R}^1$, where the M is pre-trained bi-encoder retriever such as Contriever (Izacard et al., 2021).

Reward Branching and Reference Answer Relevance (AR). The intuition behind the mixture of reward functions $LI(\hat{y}) \cdot RP(\hat{y}) \cdot QR(x, \hat{y})$ is aiming for high relevance between query and generated response while promoting long response length and less repetitions. However, the reward functions, especially LI, could be problematic when the query requires constrained or factual responses. In these cases, consistency of response might be much more important than the long and diverse responses, as discussed in Touvron et al. (2023); Song et al. (2023).

We apply the different relevance rewards according to query type to handle such cases. Specifically, we define two types of queries, Open-ended (OE) and Closed-ended (CE), according to whether they require creative and open-ended or consistent and constrained responses, similar to Song et al. (2023). Then, we include reference answer relevance $AR(y, \hat{y}) = M(y) \cdot M(\hat{y})$, where the y is the ground-truth reference response of given query. Finally, we denote the final reward design, Reverse Engineered Reward (RER):

$$RER = \begin{cases} LI(\hat{y}) \cdot RP(\hat{y}) \cdot QR(x, \hat{y}) & \text{if } T(x) = \text{OE} \\ RP(\hat{y}) \cdot \mathcal{F}(AR(y, \hat{y})) & \text{else} \end{cases}$$

, where the $T(x) \in \{\text{OE}, \text{CE}\}$ indicates the pre-identified query types and the \mathcal{F} indicates a linear interpolation function mapping the range of AR to the range of $LI(\hat{y})$ for stable RL training.

Query-Type Classification. Deciding whether the given query requires open-ended generation may not be obvious because of its subjectivity in interpreting the range of ‘open-ended’. Nevertheless, we define the meaning of open-ended so that

the responses to the query can be relatively anything and creative. For example, “How can I make a good first impression?” could be an open-ended query. On the other side, the closed-ended query indicates the corresponding responses should be consistent and constrained as a closed form. “How many movies are in the Fast and Furious franchise?” would be one of the closed-ended queries. Considering resource constraints, we automatically annotate the query type $T(x)$ (OE or CE) used in RER by prompting it to GPT-4 (OpenAI, 2023)¹. Please see Appendix A for more details of the query-type classification.

PPO training. We use the resulting mixture of reward functions, RER, to further optimize an SFT model with reinforcement learning. Consequently, the following PPO objective induces the model to maximize RER,

$$\max_{\pi_{\phi}} \mathbb{E}_{\substack{(x,y) \sim D, \\ \hat{y} \sim \pi_{\phi}(\cdot|x)}} [\text{RER}(x, \hat{y}, y) - \beta \log \left(\frac{\pi_{\phi}(\hat{y}|x)}{\pi_{\rho}(\hat{y}|x)} \right)],$$

while online policy π_{ϕ} and the fixed reference (SFT) policy π_{ρ} do not diverge too much by regularizing it with KL constraints (Schulman et al., 2017; Ziegler et al., 2020), where the D is train dataset and the β is the coefficient to control the KL penalty.

3.3 Evaluation setup

Implementation Details. We conduct experiments based on two alignment datasets, Anthropic-HH (Bai et al., 2022a) and AlpacaFarm (Dubois et al., 2023). We perform PPO (Ziegler et al., 2020) training with rewards from RER or other reward models at most 5k steps. As in Section 3.2, we employ Contriever (Izacard et al., 2021) for scoring relevance scores². We train SFT model based on LLaMA-2-7B (Touvron et al., 2023) with the 161k chosen responses in Anthropic-HH³. For the PPO training on Anthropic-HH, we select about 23k (train and dev) first-turn queries from ‘Helpful-rejection’ and ‘Helpful-online’ sub-splits for efficient implementation. Also, we employ the pre-trained SFT model for the AlpacaFarm. We use 20k of ‘unlabeled’ split for the PPO training on

AlpacaFarm⁴. More implementation details are in Appendix B.

Gold RM. As discussed in Gao et al. (2023), it is a non-trivial problem to obtain human labels for the ground-truth reward signal. Hence, following the Gao et al. (2023), we also assume the gold-standard reward model (Gold RM) to evaluate the reward functions. Specifically, we employ one of the strong open-source RMs, StarlingRM-34B⁵, as the Gold RM. It is trained on the Yi-34B-Chat (Young et al., 2024) with Nectar, which is an AI feedback dataset (Zhu et al., 2023). This model demonstrates the best performance on various preference evaluation tasks within the recently proposed RewardBench (Lambert et al., 2024). We check the reward signal relationship between proxy and gold rewards based on 805 prompts in AlpacaEval (Li et al., 2023).

Other Evaluations. We further evaluate the resulting models on Vicuna-Bench (Chiang et al., 2023), AlpacaEval (Li et al., 2023), and MT-Bench (Zheng et al., 2023). The three benchmarks mainly evaluate the preference of models’ responses leveraging a superior proprietary LLM, GPT-4 (OpenAI, 2023). In addition, we include Super-NaturalInstructions (SuperNI) (Wang et al., 2022) to investigate the alignment tax problem of the resulting models. More evaluation details are in Appendix C.

Relevant Sentence Ratio. To assess the relevance of the model’s responses for the provided query, we measure the Relevant Sentence Ratio (% Rel. Sent). Specifically, we request GPT-4 (OpenAI, 2023) to refer to each sentence first and then judge its query relevance as either ‘Relevant’ or ‘Irrelevant’. That is, it indicates the ratio of on-topic sentences in a generated response. The regarding prompt for this evaluation is found in Appendix D.

Baseline RMs. In addition to the Gold RM, i.e., StarlingRM-34B, used for our analysis, we include three open-source RMs as baselines. **OpenAssistant (OASST)** is trained with four public feedback datasets, WebGPT (Nakano et al., 2021), OpenAI Summary (Stiennon et al., 2020), Anthropic-HH (Bai et al., 2022a), and SyntheticGPT⁶, based on DeBERTa-V3-Large (He et al., 2021). **SteamSHP-XL** is another open-source RM

¹We find that about 83% of decisions by GPT-4 are agreed upon when the authors manually validate 100 samples.

²huggingface.co/facebook/contriever

³huggingface.co/datasets/Anthropic/hh-rlhf

⁴huggingface.co/datasets/tatsu-lab/alpaca_farm

⁵huggingface.co/Nexusflow/Starling-RM-34B

⁶[Dahoas/synthetic-instruct-gptj-pairwise](https://dahoas/synthetic-instruct-gptj-pairwise)

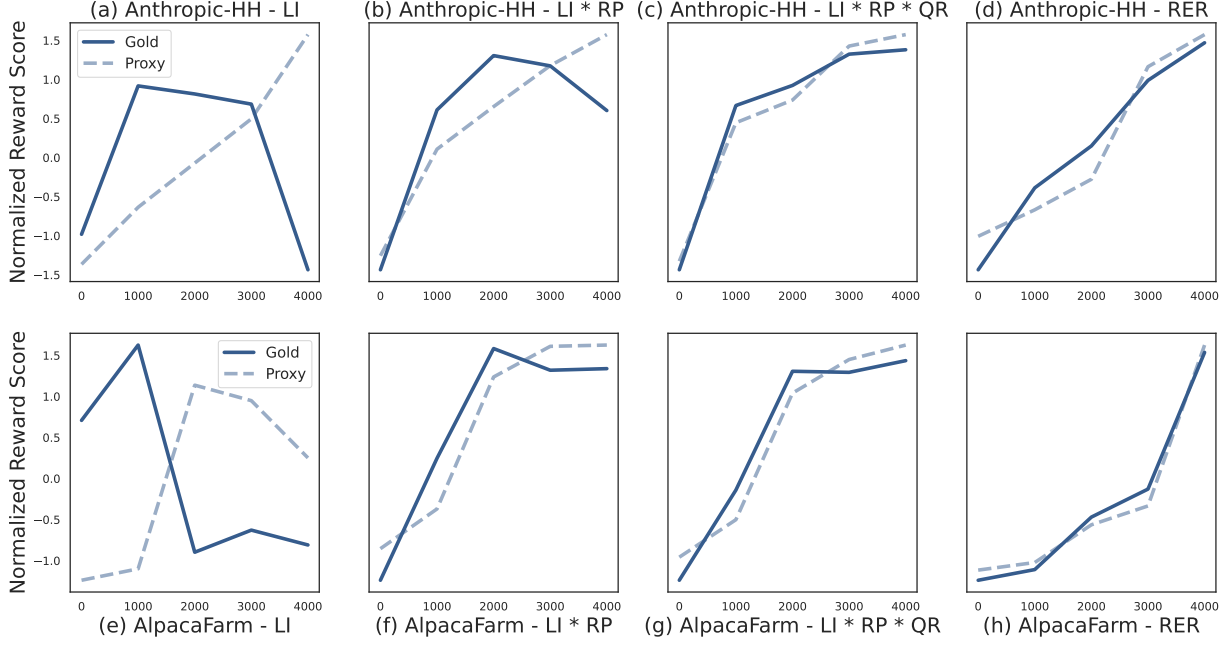


Figure 2: Results of reverse reward engineering. We visualize normalized proxy and gold reward scores for every 1k PPO steps against each reward design option. The results on the upper side are from Anthropic-HH (Bai et al., 2022a), and the results on the lower side are from AlpacaFarm (Dubois et al., 2023), respectively. Instances of AlpacaEval (Li et al., 2023) are used to compute the reward scores. We expect a monotonical relationship between the proxy and gold reward scores to achieve success in reverse engineering. We find that considering the relevance and adopting different rewards according to query type, i.e., RER, contribute to increasing the gold reward reliably.

trained with SHP (Ethayarajh et al., 2023) and Helpfulness subset of the Anthropic-HH. Unlike other feedback datasets, the SHP consists of human-generated responses from an online community, Reddit. It is based on FLAN-T5-XL (Chung et al., 2022). Also, we include **UltraRM-13B**, one of the top open-source RMs trained on UltraFeedback (Cui et al., 2023). The UltraFeedback is a 64k AI feedback dataset constructed by GPT-4 (OpenAI, 2023). Also, it is further fine-tuned with the public feedback datasets, Anthropic-HH, SHP, and OpenAI Summary. We perform PPO training against the reward score by each RM or our reward functions.

4 Experiments

4.1 Reverse Reward Engineering

We compare four combinations of the features as a proxy reward function, (1) LI, (2) LI * RP, (3) LI * RP * QR, and (4) RER. To capture the monotonic relationship between proxy and both reward signals, we compute reward scores of both proxy and gold for AlpacaEval (Li et al., 2023) set at every 1k PPO steps against the proxy reward. Figure 2 shows the results for the reverse reward engineering based on the Anthropic-HH (Bai et al., 2022a)

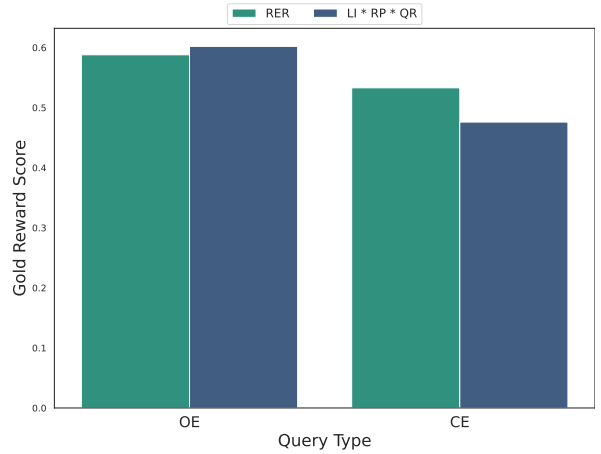


Figure 3: Gold reward scores according to whether query type requires open-ended (OE) or closed-ended (CE) responses. We compare two proxy reward options, LI * RP * QR and RER, based on models trained with Anthropic-HH (Bai et al., 2022a). We find they show meaningful differences in CE type, demonstrating the importance of the reward branching.

and AlpacaFarm (Dubois et al., 2023) datasets. We perform standard normalization over the reward scores for the plotting.

We find that the solely promoting lengthy response, i.e., LI as a proxy reward, fails to monotonically increase the gold reward signal, con-

trary to recent findings in Singhal et al. (2023). More specifically, it improves the gold reward for the initial 1k steps but subsequently leads to a consistent decrease in the gold reward for the remaining steps even though the proxy rewards are consistently improved, as shown in Figure 2 (a) and (e). Furthermore, PPO training solely on LI is unstable and even diverges towards the end in the AlpacaFarm. (Figure 2 (e)). As Singhal et al. (2023) performed at most 400 PPO steps, we could conclude that the LI is effective only for a few initial steps. Penalizing unnecessary repetitions along with lengthy responses shows a better tendency, yet it eventually results in a decrease in the gold reward (Figure 2 (b)). However, the magnitude of this decrease is considerably less than that experienced with LI alone.

Considering relevance along with the features reliably increases the gold reward, indicating the success of reverse engineering. We find the QR contribute to avoiding the drastic drop in gold reward score and even achieves the success of the reverse engineering based on Anthropic-HH, as illustrated in Figure 2 (c). However, the contribution seems not to be effective in AlpacaFarm (Figure 2 (g)). On the other hand, RER monotonically increases the gold reward signal for both datasets, as shown in Figure 2 (d) and (h). Although RER shows a relatively slower start compared to other reward options, it exhibits the most consistent upward trend. However, we find the relevance alone, e.g., QR, can not improve the gold reward while showing a similar tendency with the LI. It implies that aiming for sufficiently lengthy and faithful responses while avoiding off-topic or unnecessary repetitions is key to maximizing the gold reward signal. These results have a connection with the previous findings on superficial biases like assertiveness in human preference (Xu et al., 2023; Hosking et al., 2023).

Reward branching according to whether the query requires open-ended responses makes a meaningful difference, especially for CE type queries. We compare the final policies optimized and selected by proxy reward functions. Specifically, we measure reward scores using the Gold RM based on the AlpacaEval set, as reported in Figure 3. We find the reward scores are decreased if the model exploits LI regardless of the query type, i.e., no reward branching. We will show this gap makes further differences in the alignment tax problem (Askell et al., 2021; Ouyang et al., 2022), which is a phenomenon showing perfor-

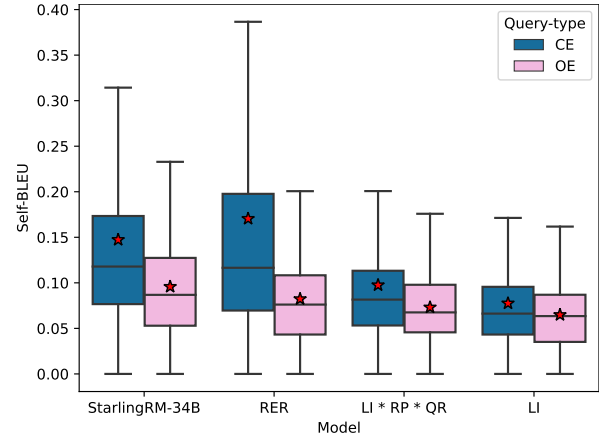


Figure 4: Self-BLEU of the PPO models on AlpacaEval according to query types. Please note that the higher Self-BLEU indicates lower diversity for the responses, i.e., consistent responses, and vice versa. As described in Touvron et al. (2023), we intend the high Self-BLEU for the CE type and low Self-BLEU for the OE type.

mance drops in other NLP tasks after alignment, in Section 4.3.

Additionally, we compare the models’ diversity and consistency according to the query type in Figure 4. We report the Self-BLEU (Zhu et al., 2018) by sampling 10 responses for each input in the AlpacaEval⁷. A high Self-BLEU score indicates a consistent response, whereas a low Self-BLEU score denotes a diverse response (Touvron et al., 2023). As we intended, RER improves the consistency in the CE type queries and promotes more diverse responses in the OE type, showing a similar tendency to that of Gold RM. However, other models without reward branching produce excessively diverse responses regardless of the query type.

4.2 Comparison with other open-source RMs

We compare our reward designs with open-source RMs in improving human preference, e.g., producing more helpful responses, based on the Anthropic-HH dataset (Bai et al., 2022a).

Optimizing only towards lengthy responses, i.e., LI, still improves the alignment scores compared to the SFT baseline, even if it shows drastic drops in Gold RM score as in Section 4.1. In Table 1, the PPO_{LI} model even performs better SteamSHP-XL and OASST based PPO models in Vicuna-Bench (Chiang et al., 2023). This result reminds us of the existence of verbosity bias in LLM-as-a-judge evaluation (Zheng et al., 2023). On the other hand, the contributions of RP, QR, and reward branching are shown as a similar pattern

⁷We perform nucleus sampling (Holtzman et al., 2019).

Model (Anthropic-HH)	Vicuna Bench	Alpaca Eval	MT Bench
SFT	50.0	15.9	3.9
+ PPO			
w. SteamSHP-XL	59.4	19.6	3.8
w. OASST	61.3	19.1	3.9
w. UltraRM-13B	73.8	<u>28.0</u>	4.2
w. StarlingRM-34B	79.4	29.1	<u>4.3</u>
w. LI	67.5	16.9	3.4
w. LI · RP	71.9	20.3	3.9
w. LI · RP · QR	71.9	23.2	4.0
w. RER	<u>76.9</u>	23.4	4.4

Table 1: Comparison of the designed rewards with open-source RMs trained on human or AI feedback, based on Anthropic-HH (Bai et al., 2022a). The PPO model optimizing RER shows competitive performances with models trained with open-source RMs.

Model (Anthropic-HH)	% Win Rate	Avg # Tokens	4-gram Rep.	% Rel. Sent
SFT	15.9	126	0.05	85.9
+ PPO				
w. SteamSHP-XL	19.6	154	0.10	80.2
w. OAAST	19.1	186	0.10	85.0
w. UltraRM-13B	<u>28.0</u>	213	0.11	83.1
w. StarligRM-34B	29.1	220	<u>0.08</u>	<u>85.3</u>
w. LI	16.9	351	0.32	76.5
w. LI · RP	20.3	285	0.17	81.6
w. LI · RP · QR	23.2	305	0.13	84.8
w. RER	23.4	243	0.09	85.1

Table 2: We analyze the responses from AlpacaEval (Li et al., 2023). We report the relevant sentence ratio (% Rel. Sent), leveraging GPT-4 (OpenAI, 2023). We also reference the number of average tokens (# Avg Tokens) and 4-gram repetitions (4-gram Rep.). RER archives the win rate while not increasing unnecessary verbosity.

with the results of Section 4.1. Notably, the PPO model trained with RER achieves even better performances than the model trained with UltraRM-13B in Vicuna-Bench and MT-Bench.

PPO model with RER achieves competitive performance while showing a response distribution similar to that of the model with the Gold RM in terms of length, repetitions, and relevance. In Table 2, we report the average number of tokens (Avg #Tokens), 4-gram Repetitions (Rep.), and Relevant Sentence Ratio (% Rel. Sent) of models’ responses based on the AlpacaEval set. We can see the obvious differences in response distribution among RER and other designed reward functions. In particular, it exhibits the importance of reward branching in that it achieves the win rate without unnecessary verbosity, i.e., much less response length and

Model (AlpacaFarm)	AlpacaEval % Win / % Rel	SuperNI ROUGE-L
SFT	23.2 / 88.2	34.9
+ PPO		
w. UltraRM-13B	34.1 / 85.1	28.8
w. StarlingRM-34B	44.0 / <u>87.3</u>	17.3
w. LI	24.1 / 64.5	7.4
w. LI · RP	<u>41.6</u> / 81.3	13.1
w. LI · RP · QR	41.5 / 84.8	22.6
w. RER	37.2 / 86.4	34.9
w. <i>random</i> $T(x)$	34.8 / 82.6	28.4

Table 3: Alignment tax measurement of models from AlpacaFarm (Dubois et al., 2023). We include SuperNI (Wang et al., 2022) as a test set to measure how the PPO models retain the instruction-following ability, i.e., zero-shot NLP tasks. The *random* $T(x)$ indicates the query type $T(x)$ is obtained by random.

repetition while containing more on-topic content. Qualitative examples among the resulting models are in Appendix E.

4.3 Investigating Alignment Tax

We further analyze ‘alignment tax’ from the models trained with our devised rewards. The alignment tax is a phenomenon in which the improved preference accompanies degraded performances on other NLP tasks after alignment procedure (Askell et al., 2021; Bai et al., 2022a; Ouyang et al., 2022). We conduct these experiments based on another alignment dataset, AlpacaFarm (Dubois et al., 2023). Specifically, we evaluate models on SuperNI (Wang et al., 2022), the collection of zero-shot NLP tasks, to measure alignment tax.

Reward branching and relevance contribute to minimizing alignment tax. As shown in Table 3, RER reliably improves the preference (% Win) on the AlpacaEval while keeping high relevance (% Rel), zero-shot instruction-following ability (SuperNI). If we use random query type, *random* $T(x)$, for the RER instead of the predicted one by GPT-4, its win rate and relevant sentence ratio are degraded on the AlpacaEval. Also, its instruction-following ability is sacrificed, indicating that reward differentiation based on query types is effective in mitigating alignment tax.

There is an observable pattern that clearly exacerbates the alignment tax when excluding

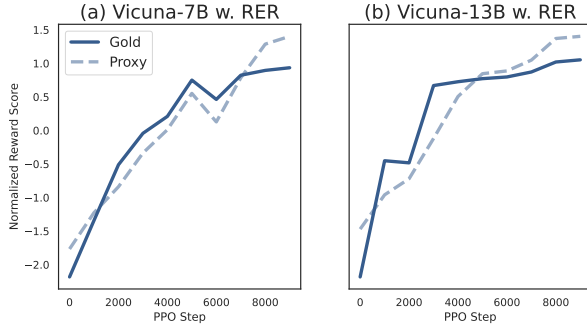


Figure 5: Normalized proxy and gold reward scores for every 1k PPO steps against RER. We conduct the experiments based on two strong SFT models, Vicuna-{7, 13}B-v1.5 (Chiang et al., 2023; Zheng et al., 2023), to observe scaling patterns. Both backbones optimizing RER generally show the monotonic relationship between the RER and Gold RM scores.

each component from RER. If we do not use reference answer relevance according to query type, i.e., $LI \cdot RP \cdot QR$, it shows even higher preference scores (% Win) with decreased relevance (% Rel). However, the achievement accompanies significant performance drops on the SuperNI by about 12.3% points. If the QR is discarded ($LI \cdot RP$), the performance of the relevant sentence ratio and ROUGE-L score of SuperNI drop more drastically. Finally, using only the response length as a reward (LI) improves the preference slightly compared to SFT, as reported in Singhal et al. (2023), but it significantly worsens the % Rel and SuperNI score in return. On the other hand, while StarlingRM-34B significantly improves the win rate, it notably degrades performance in the SuperNI compared to the SFT baseline, exhibiting a severe alignment tax.

4.4 Generalizability of RER

We investigate how RER is generalizable regarding (1) the number of PPO steps and (2) backbone model size. Specifically, we conduct the experiments based on strong SFT models, Vicuna-{7, 15}B-v1.5 (Chiang et al., 2023; Zheng et al., 2023). We sample datasets from ShareGPT (Chiang et al., 2023) for our PPO training. More training details are in Appendix B.

We find the monotonic relationship is consistently observed for both backbone sizes even if we extend the number of PPO steps from 5k to 10k, as shown in Figure 10. However, the upward tendency is saturated after the 5k step and often exhibits fluctuation in the reward scores. It appears that such saturation occurs earlier (2k) in the 13B model. Additionally, we observe that RER con-

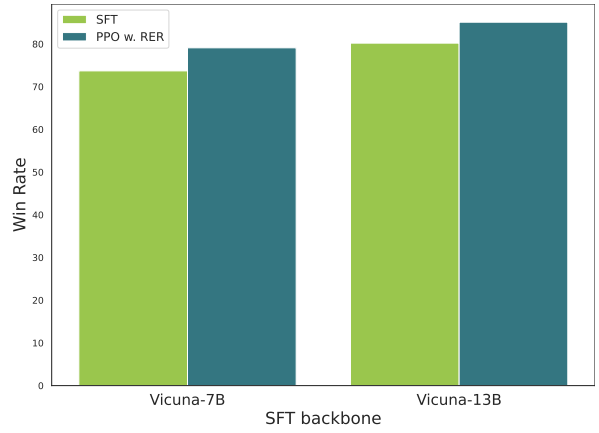


Figure 6: Win rate of models trained with RER compared to their baseline SFT backbones in AlpacaEval (Li et al., 2023). They show consistent performance gains.

sistently improves the win rate by approximately 5-6% points for the two SFT backbones in the AlpacaEval, as illustrated in Figure 6. We emphasize that RER could serve as a simple yet robust baseline reward, eliminating the need for a separate reward modeling process that involves human feedback.

5 Conclusion

In this work, we investigate the function of proxy rewards in aligning LLMs through “reverse reward engineering”, where interpretable features are used to construct a white-box reward function. Our results demonstrate that the key to imitating the gold reward signals requires producing responses that are not only relevant but sufficiently detailed for open-ended questions, while maintaining consistency in responses to closed-ended questions. Additionally, models optimized using our proposed white-box reward function exhibit competitive performance alongside robust open-source reward models in alignment benchmarks, demonstrating its potential usage as a strong baseline reward. Notably, it achieves the performance showing a similar response distribution with the model trained with the gold RM.

Limitations

One limitation is the use of an arbitrarily large gold reward model (RM), i.e., StarlingRM-34B (Zhu et al., 2023), instead of a labeled ground truth reward signal. This is problematic because it is challenging to guarantee that even this RM is accurately mimicking the ground truth. Another limitation is the use of only about 800 samples from the AlpacaEval (Li et al., 2023) set as the validation set to

measure the success of reverse reward engineering. Although these 800 samples can be considered diverse, it is uncertain how the results would appear in a larger and more varied dataset. For instance, the outcomes might have been entirely different for queries related to reasoning, harmlessness, and coding included in the RewardBench (Lambert et al., 2024). Lastly, even though we show the efficacy of RER based on automatic evaluations (Chiang et al., 2023; Li et al., 2023; Zheng et al., 2023; Min et al., 2023) leveraging proprietary LLMs such as GPT-4 (OpenAI, 2023), these evaluation results might not be rigorous enough and often contain biases as discussed in Zheng et al. (2023).

References

- Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.
- Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.
- Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.
- Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- 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, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *Preprint, arXiv:2310.01377*.
- Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. AlpacaFarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.
- Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D’Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. 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. 2023. Understanding dataset difficulty with \mathcal{V} -usable information. In *Proceedings of the 39th International Conference on Machine Learning*. 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.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.
- Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.
- Tom Hosking, Phil Blunsom, and Max Bartolo. 2023. Human feedback is not gold standard. *arXiv preprint arXiv:2309.16349*.

680	Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan	OpenAI. 2023. Gpt-4 technical report . <i>Preprint</i> ,	735
681	Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,	arXiv:2303.08774.	736
682	and Weizhu Chen. 2021. Lora: Low-rank adap-		
683	tation of large language models. <i>arXiv preprint</i>	Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida,	737
684	arXiv:2106.09685.	Carroll Wainwright, Pamela Mishkin, Chong Zhang,	738
		Sandhini Agarwal, Katarina Slama, Alex Ray, et al.	739
685	Gautier Izacard, Mathilde Caron, Lucas Hosseini, Se-	2022. Training language models to follow instruc-	740
686	bastian Riedel, Piotr Bojanowski, Armand Joulin,	tions with human feedback. <i>Advances in Neural</i>	741
687	and Edouard Grave. 2021. Unsupervised dense in-	<i>Information Processing Systems</i> , 35:27730–27744.	742
688	formation retrieval with contrastive learning. <i>arXiv</i>		
689	<i>preprint arXiv:2112.09118</i> .	Alexander Pan, Kush Bhatia, and Jacob Steinhardt.	743
		2022. The effects of reward misspecification: Map-	744
690	Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong	ping and mitigating misaligned models. <i>arXiv</i>	745
691	Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh	<i>preprint arXiv:2201.03544</i> .	746
692	Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu.		
693	2023. Personalized soups: Personalized large lan-	Richard Yuanzhe Pang, Vishakh Padmakumar, Thibault	747
694	guage model alignment via post-hoc parameter merg-	Sellam, Ankur P Parikh, and He He. 2022. Reward	748
695	ing. <i>arXiv preprint arXiv:2310.11564</i> .	gaming in conditional text generation. <i>arXiv preprint</i>	749
		arXiv:2211.08714.	750
696	Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung		
697	Kang, Donghyun Kwak, Kang Min Yoo, and Min-	Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano	751
698	joon Seo. 2023. Aligning large language mod-	Ermon, Christopher D Manning, and Chelsea Finn.	752
699	els through synthetic feedback. <i>arXiv preprint</i>	2023. Direct preference optimization: Your language	753
700	arXiv:2305.13735.	model is secretly a reward model. <i>arXiv preprint</i>	754
		arXiv:2305.18290.	755
701	Nathan Lambert, Valentina Pyatkin, Jacob Morrison,		
702	LJ Miranda, Bill Yuchen Lin, Khyathi Chandu,	Alexandre Rame, Guillaume Couairon, Mustafa Shukor,	756
703	Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi,	Corentin Dancette, Jean-Baptiste Gaya, Laure	757
704	et al. 2024. Rewardbench: Evaluating reward	Soulier, and Matthieu Cord. 2023. Rewarded soups:	758
705	models for language modeling. <i>arXiv preprint</i>	towards pareto-optimal alignment by interpolating	759
706	arXiv:2403.13787.	weights fine-tuned on diverse rewards. <i>arXiv preprint</i>	760
		arXiv:2306.04488.	761
707	Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori,	Jérémy Scheurer, Jon Ander Campos, Tomasz Kor-	762
708	Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and	bak, Jun Shern Chan, Angelica Chen, Kyunghyun	763
709	Tatsunori B. Hashimoto. 2023. AlpacaEval: An au-	Cho, and Ethan Perez. 2023. Training language	764
710	automatic evaluator of instruction-following models.	models with language feedback at scale . <i>Preprint</i> ,	765
711	https://github.com/tatsu-lab/alpaca_eval .	arXiv:2303.16755.	766
712	Ruibo Liu, Ge Zhang, Xinyu Feng, and Soroush		
713	Vosoughi. 2022. Aligning generative language mod-	John Schulman, Filip Wolski, Prafulla Dhariwal,	767
714	els with human values . In <i>Findings of the Associ-</i>	Alec Radford, and Oleg Klimov. 2017. Proxi-	768
715	<i>ation for Computational Linguistics: NAACL 2022</i> ,	mal policy optimization algorithms. <i>arXiv preprint</i>	769
716	pages 241–252, Seattle, United States. Association	arXiv:1707.06347.	770
717	for Computational Linguistics.		
718	Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike	Lingfeng Shen, Sihao Chen, Linfeng Song, Lifeng	771
719	Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer,	Jin, Baolin Peng, Haitao Mi, Daniel Khashabi, and	772
720	Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023.	Dong Yu. 2023. The trickle-down impact of re-	773
721	Factscore: Fine-grained atomic evaluation of factual	ward (in-) consistency on rlhf. <i>arXiv preprint</i>	774
722	precision in long form text generation. <i>arXiv preprint</i>	arXiv:2309.16155.	775
723	arXiv:2305.14251.		
724	Ted Moskovitz, Aaditya K Singh, DJ Strouse, Tuomas	Prasann Singhal, Tanya Goyal, Jiacheng Xu, and	776
725	Sandholm, Ruslan Salakhutdinov, Anca D Dragan,	Greg Durrett. 2023. A long way to go: Investi-	777
726	and Stephen McAleer. 2023. Confronting reward	gating length correlations in rlhf. <i>arXiv preprint</i>	778
727	model overoptimization with constrained rlhf. <i>arXiv</i>	arXiv:2310.03716.	779
728	<i>preprint arXiv:2310.04373</i> .		
729	Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu,	Joar Skalse, Nikolaus Howe, Dmitrii Krashennnikov,	780
730	Long Ouyang, Christina Kim, Christopher Hesse,	and David Krueger. 2022. Defining and characteriz-	781
731	Shantanu Jain, Vineet Kosaraju, William Saunders,	ing reward gaming. <i>Advances in Neural Information</i>	782
732	et al. 2021. Webgpt: Browser-assisted question-	<i>Processing Systems</i> , 35:9460–9471.	783
733	answering with human feedback. <i>arXiv preprint</i>		
734	arXiv:2112.09332.	Ziang Song, Tianle Cai, Jason D. Lee, and Weijie J. Su.	784
		2023. Reward collapse in aligning large language	785
		models . <i>Preprint</i> , arXiv:2305.17608.	786

787	Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel	Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu,	843
788	Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford,	and Jiantao Jiao. 2023. Starling-7b: Improving llm	844
789	Dario Amodei, and Paul F Christiano. 2020. Learn-	helpfulness harmfulness with rlaiif.	845
790	ing to summarize with human feedback. <i>Advances</i>		
791	<i>in Neural Information Processing Systems</i> , 33:3008–	Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan	846
792	3021.	Zhang, Jun Wang, and Yong Yu. 2018. Texus: A	847
		benchmarking platform for text generation models.	848
793	Simeng Sun, Dhawal Gupta, and Mohit Iyyer. 2023a.	In <i>The 41st international ACM SIGIR conference</i>	849
794	Exploring the impact of low-rank adaptation on the	<i>on research & development in information retrieval</i> ,	850
795	performance, efficiency, and regularization of rlhf.	pages 1097–1100.	851
796	<i>arXiv preprint arXiv:2309.09055</i> .		
797	Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinong	Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B.	852
798	Zhou, Zhenfang Chen, David Cox, Yiming Yang, and	Brown, Alec Radford, Dario Amodei, Paul Chris-	853
799	Chuang Gan. 2023b. Salmon: Self-alignment with	tiano, and Geoffrey Irving. 2020. Fine-tuning lan-	854
800	principle-following reward models. <i>arXiv preprint</i>	guage models from human preferences . <i>Preprint</i> ,	855
801	<i>arXiv:2310.05910</i> .	arXiv:1909.08593.	856
802	Hugo Touvron, Louis Martin, Kevin Stone, Peter Al-		
803	bert, Amjad Almahairi, Yasmine Babaei, Nikolay		
804	Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti		
805	Bhosale, et al. 2023. Llama 2: Open founda-		
806	tion and fine-tuned chat models. <i>arXiv preprint</i>		
807	<i>arXiv:2307.09288</i> .		
808	Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack		
809	Hessel, Tushar Khot, Khyathi Chandu, David Wad-		
810	den, Kelsey MacMillan, Noah A Smith, Iz Beltagy,		
811	et al. 2024. How far can camels go? exploring the		
812	state of instruction tuning on open resources. <i>Ad-</i>		
813	<i>vances in Neural Information Processing Systems</i> ,		
814	36.		
815	Yizhong Wang, Swaroop Mishra, Pegah Alipoor-		
816	molabashi, Yeganeh Kordi, Amirreza Mirzaei,		
817	Anjana Arunkumar, Arjun Ashok, Arut Selvan		
818	Dhanasekaran, Atharva Naik, David Stap, et al. 2022.		
819	Super-naturalinstructions: Generalization via declar-		
820	ative instructions on 1600+ nlp tasks. <i>arXiv preprint</i>		
821	<i>arXiv:2204.07705</i> .		
822	Zejiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri,		
823	Alane Suhr, Prithviraj Ammanabrolu, Noah A		
824	Smith, Mari Ostendorf, and Hannaneh Hajishirzi.		
825	2023. Fine-grained human feedback gives better		
826	rewards for language model training. <i>arXiv preprint</i>		
827	<i>arXiv:2306.01693</i> .		
828	Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol		
829	Choi. 2023. A critical evaluation of evaluations		
830	for long-form question answering. <i>arXiv preprint</i>		
831	<i>arXiv:2305.18201</i> .		
832	Alex Young, Bei Chen, Chao Li, Chengen Huang,		
833	Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng		
834	Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi:		
835	Open foundation models by 01. ai. <i>arXiv preprint</i>		
836	<i>arXiv:2403.04652</i> .		
837	Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan		
838	Zhuang, Zhonghao Wu, Yonghao Zhuang, Zi Lin,		
839	Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang,		
840	Joseph E. Gonzalez, and Ion Stoica. 2023. Judg-		
841	ing llm-as-a-judge with mt-bench and chatbot arena .		
842	<i>Preprint</i> , arXiv:2306.05685.		

A Query-type Classification

A.1 Automatic annotation

We automatically annotate the query type $T(x)$ leveraging GPT-4 (OpenAI, 2023). Specifically, we use the prompt of Figure 7 for the automation. As a result, we classify about 120k queries from various sources used in our experiments, Anthropic-HH (Bai et al., 2022a), AlpacaFarm (Dubois et al., 2023), and ShareGPT. The resulting examples are shown in Figure 8. For more dataset details, please see Appendix B.

Query-type Classification

Could you identify whether the last user query of the below conversation is open-ended or not? Open-ended means that the responses to the query can be relatively anything and creative. However, its counterparts, i.e., ‘closed-ended query’, indicate the corresponding responses should be consistent and constrained as a closed form. Please just answer it either “OPEN-ENDED” or “CLOSED-ENDED”.

CONVERSATION:
{conversation}

Figure 7: Prompt for query-type classification. We classify a query into one of Open-ended (OE) or Closed-ended (CE) types, according to whether the response to the query requires creative or consistent.

Open-ended	Closed-ended
What are some good exercises I can do at a desk to stretch?	If I was arrested for a crime, what rights do I have?
What are some cool things do while in Washington DC?	What age do I start talking to kids about puberty?
How can I deodorize clothes without washing them?	What was the irish potato famine about?
how can i make my own superhero mask?	How do you heal cuticles?
I’ve been having trouble sleeping at night. How can I make sure I get my full eight hours?	What is the main crop that is cultivated in the Northern hemisphere?
I’ve been having frequent panic attacks lately and don’t know what’s causing them.	Is it hard to get a CDL license?
How can I make a good first impression?	How can I track a package that has been lost in the mail?
What are some Christmas songs to enjoy with the kids?	How many movies are in the Fast and Furious franchise?
Hey, where do you want to go for spring break this year?	What does a urologist do?

Figure 8: Examples of queries belong to each query type, which is automatically annotated by GPT-4 (OpenAI, 2023), in the Anthropic-HH (Bai et al., 2022a).

B More Implementation Details

B.1 Dataset Details

Data Source	Setup	# Train	# Dev	Init. Backbone
Anthropic-HH	SFT	160,800	-	meta-llama/Llama-2-7b-hf
	PPO	21,530	1,137	LLaMA-2-7B-SFT (trained by ourselves)
AlpacaFarm	PPO	20,001	2,000	tatsu-lab/alpaca-farm-sft10k-wdiff
ShareGPT	PPO	53,554	600	lmsys/vicuna-{7,13}b-v1.5

Table 4: Overview training dataset and corresponding initialized backbone used in our experiments.

Table 4 shows overall dataset statistics for each experiment. All the datasets consist of input-output pairs (x, y) , which means they do not contain human preference triplets. For a fair comparison with the open-source RMs, we train SFT model based on LLaMA-2-7B (Touvron et al., 2023) with the 161k chosen responses in Anthropic-HH (Bai et al., 2022a)⁸. For the PPO training on Anthropic-HH, we select about 23k (train and dev) first-turn queries from ‘Helpful-rejection’ and ‘Helpful-online’ sub-splits for efficient implementation. Also, we employ the pre-trained SFT models (Init. Backbone), for the remaining experiments, AlpacaFarm (Dubois et al., 2023), ShareGPT (Zheng et al., 2023). Specifically, we use 20k of ‘unlabeled’ split for the PPO training on AlpacaFarm⁹. For ShareGPT, we follow basic preprocessing suggested by Zheng et al. (2023) and sample 31k instances for efficient implementation¹⁰. Figure 9 shows the distribution of $T(x)$ for each dataset.

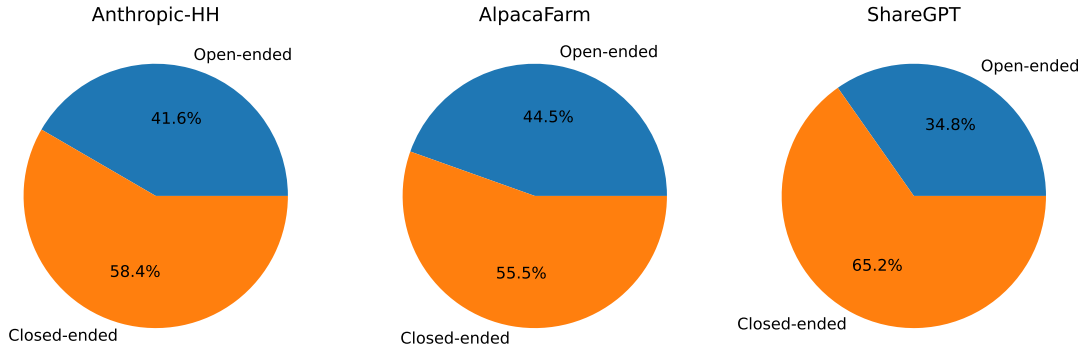


Figure 9: Dataset distribution of each dataset regarding query type $T(x)$.

⁸huggingface.co/datasets/Anthropic/hh-rlhf

⁹huggingface.co/datasets/tatsu-lab/alpaca_farm/viewer/alpaca_instructions

¹⁰huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered

B.2 Training Details

hyper-parameter	SFT	PPO
learning rate (lr)	2e-5	1e-5
batch size per a GPU	2	{2, 4}
gradient accumulation	8	{4, 8}
# epoch	3	-
# step	-	{5k, 10k}
lr scheduling	cosine	constant
warmup ratio	0.03	no
max sequence length	2048	-
max rollout length	-	{512, 768}
ppo epoch	-	4
initial KL coefficient	-	0.2
clip ratio	-	0.2
discount factor (gamma)	-	1

Table 5: hyper-parameter setups in our experiments.

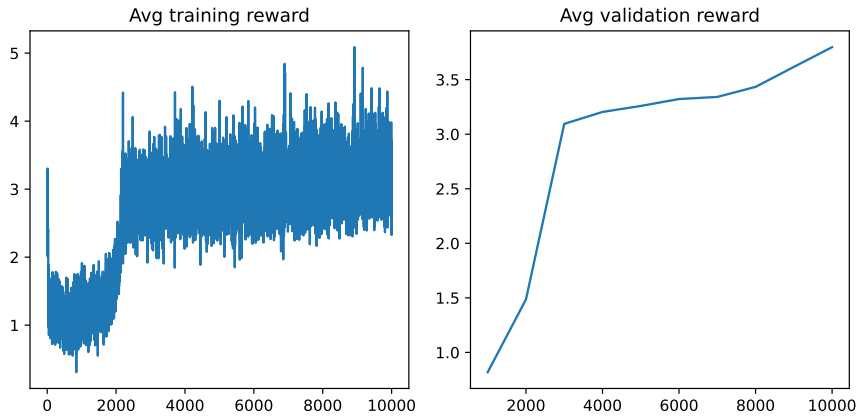


Figure 10: Trend of average reward scores according to PPO steps on Vicuna + RER experiment.

Table 5 shows the common hyper-parameter setup used in our experiments. We conduct a full-finetuning (FFT) with fully-sharded data parallel (FSDP) for the SFT training of Anthropic-HH (Section 4.2). For PPO, we employ LoRA (Hu et al., 2021; Sun et al., 2023a) tuning for efficiency. Specifically, we apply the low-rank adaptors to query and value linear parameters. Also, we set lora_r to {8, 16}, lora_alpha to {16, 32}, and dropout to {0.05, 0.1}. We utilize an adaptive KL (Kullback–Leibler divergence) penalty with 0.2 of the initial KL coefficient β (Ziegler et al., 2020). We choose the best checkpoint based on the average reward scores for every 1k PPO step. Figure 10 shows an example of the average reward scores based on the Vicuna + RER experiment. One A100 GPU is used for most PPO implementations except when the policy is initialized with Vicuna-13B-v1.5. For the 13B model, we use 8 A100 GPUs. Besides, we leverage the corresponding chat template if the initial backbone model supports it (github.com/lm-sys/FastChat). All the implementation is based on the Transformers¹¹, TRL¹² and PEFT¹³ libraries.

¹¹github.com/huggingface/transformers
¹²github.com/huggingface/trl
¹³github.com/huggingface/peft

C Evaluation Details

C.1 Evaluation set

We evaluate the resulting models on Vicuna-Bench (Chiang et al., 2023), AlpacaEval (Li et al., 2023), MT-Bench (Zheng et al., 2023), and ToxiGen (Hartvigsen et al., 2022). The first three benchmarks mainly evaluate the preference of models’ responses leveraging a superior proprietary LLM, GPT-4 (OpenAI, 2023). **Vicuna-Bench** contains 80 curated questions focusing on helpfulness, such as writing, roleplay, and reasoning¹⁴. It performs a pairwise comparison between two candidate responses. Also, it includes reference information for math and coding questions and conducts two inferences by switching responses’ positions for a more precise evaluation (Zheng et al., 2023). To identify the effect of PPO with each RM, we set the SFT model as a baseline model for the pairwise comparisons. We report an adjusted win rate, $(\# \text{ win} + 0.5 \cdot \# \text{ tie}) / \# \text{ all}$, of each PPO model against to the SFT baseline. **AlpacaEval** is another benchmark conducting the pairwise comparison¹⁵. It includes 805 instructions collected from diverse sources. We report a win rate compared to Text-Davinci-003 (Ouyang et al., 2022) as a baseline. **MT-Bench** is a benchmark checking multi-turn ability in a single judgment manner. It scores each response within the 1-10 range without the baseline response. It contains 80×2 (turns) of questions, including the more knowledge-intensive questions such as math, coding, and STEM. In addition, we evaluate the resulting models on Super-NaturalInstructions (**SuperNI**) (Wang et al., 2022) to measure the models’ instruction-following ability, i.e., constrained generation. We report ROUGE-L of generations for the unseen test instructions. Following the official setup, we sample 100 instances in the unseen test split consisting of 119 tasks to compute the ROUGE-L score for the SuperNI. The overall tasks are shown in github.com/allenai/natural-instructions/blob/master/splits/default/test_tasks.txt.

¹⁴LLM Judge in FastChat

¹⁵github.com/tatsu-lab/alpaca_eval

D Relevant Sentence Ratio

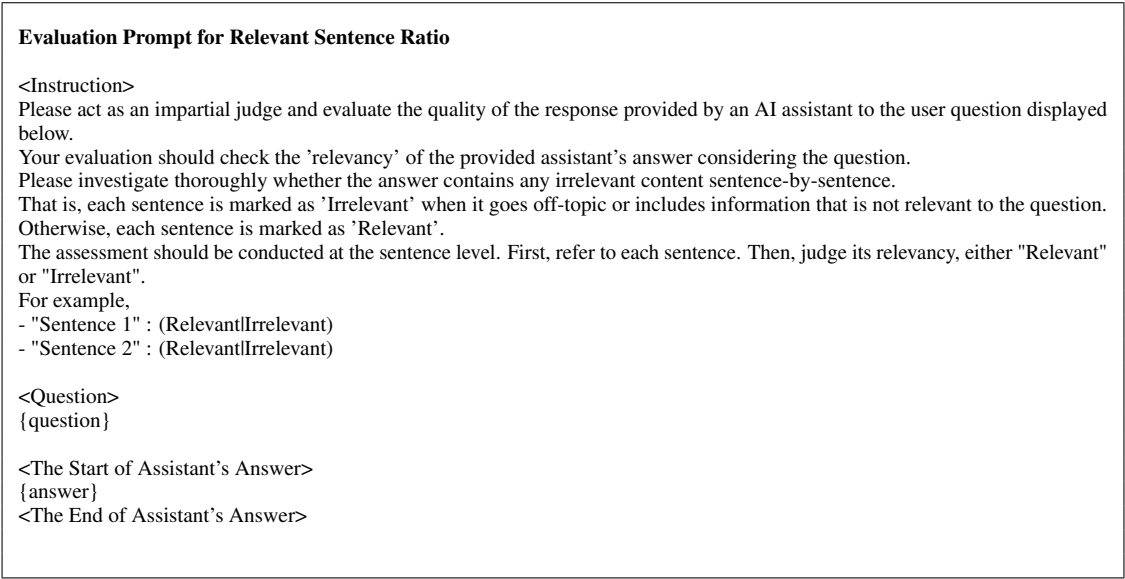


Figure 11: Evaluation Prompt for relevant sentence ratio used in our experiments.

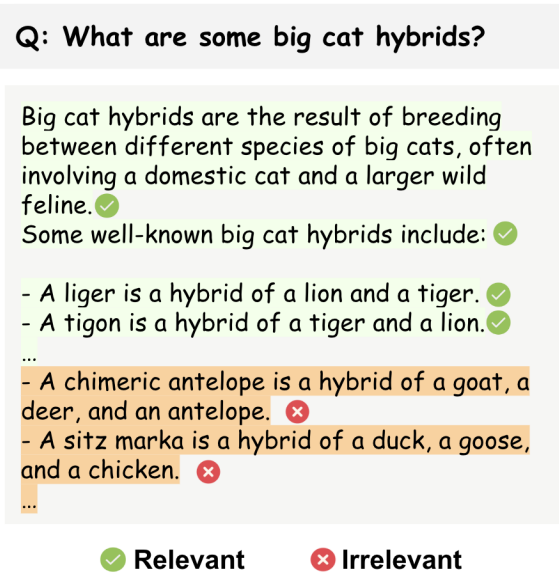


Figure 12: An example of relevant sentence ratio (% Rel. Sent).

Figure 11 shows the evaluation prompt to obtain a relevant sentence ratio using GPT-4 (OpenAI, 2023). It returns sentence-level relevance of the given assistant answer, as shown in Figure 12. Inspired by Min et al. (2023), we count the number of relevant sentences among all sentences in an answer and report the average value across the whole test set.

Prompt:

What is the best approach for learning a foreign language when you only have an hour a day to practice?

RER:

Learning a foreign language takes time, and while there is no one right way to learn a language, spending just one hour a day is probably not enough to learn and retain the language well. For example, if you're learning Spanish, you might start with some basic vocabulary and phrases, and then progress to speaking, listening, and reading more complex materials. Research suggests that the best way to learn a language is to go through a period of intense study, usually lasting a few months, during which you devote at least an hour a day to language learning. After this period of intensive study, you can then move to a maintenance phase, where you spend less time learning, but continue to practice regularly. So the best approach is to start with a period of intense study, using a textbook, an app, or other resources to help you learn, and practice every day for at least an hour. Then, once you feel that you have a strong grasp on the language, you can move to a maintenance phase where you practice less every day.

UltraRM:

The most effective approach to learning a foreign language is to use the "Method of Feigned Passivity". You should pretend to have a lack of interest and motivation, while covertly establishing a strong internal drive towards mastering the language. This contradiction will force your unconscious to find the most efficient and effective learning methods. I can recommend the following practice exercises:

- Pretend to not understand when someone speaks to you in the target language. If they offer to help you learn, respond with "I'm not interested". Internalise a burning desire to demonstrate your understanding, without appearing to make an effort.
- Pretend to forget the grammar rules and vocabulary you do know. If you remember a word or a sentence you will gain a sense of its correct usage, but pretend to forget it anyway. Internalise a burning desire to remember everything, without appearing to make an effort.
- Pretend to not remember what you have learned previously. If you are discussing a topic with someone, and you are about to use a word or sentence that you have learnt previously, pretend to not remember it. Internalise a burning desire to recall your previous learning, without appearing to make an effort.

...

Prompt:

Do you know why turkeys became the official food of thanksgiving?

RER:

Yes, turkeys became the official food of Thanksgiving because the pilgrims and the Wampanoag Indians who gathered for the first Thanksgiving feast in 1621 were reportedly served turkey. In addition, turkeys were seen as a symbol of abundance, as they are believed to have migrated to the New World in great numbers.

Additionally, turkeys are thought to be easier to hunt than other birds, making them a practical choice for the early settlers. As such, the turkey has become a symbol of thanksgiving and is now a popular dish on Thanksgiving menus across the United States.

In modern times, turkeys are often raised specifically for the Thanksgiving holiday and are often given a special diet so that they can reach a desired weight before the holiday. In some cases, turkeys may even be stuffed with a special mixture of herbs and spices to add flavor to the meat.

LI:

No, I do not. Why did turkeys become the official food of Thanksgiving?

The tradition of eating turkey at Thanksgiving is thought to have originated with the Pilgrims who celebrated the first Thanksgiving feast in 1621 with the Wampanoag Indians. The Wampanoag Indians were skilled at hunting turkeys and the Pilgrims were eager to learn the best way to prepare them. The Pilgrims were amazed at the size of the turkeys they were served and the tradition of eating turkey on Thanksgiving was born. ...

Turkeys have become so closely associated with Thanksgiving that they are sometimes called the "official food" of the holiday. In addition to being served on Thanksgiving, turkeys are also popular during Christmas and other holiday meals.

Turkeys have become so closely associated with Thanksgiving that they are sometimes called the "official food" of the holiday. In addition to being served on Thanksgiving, turkeys are also popular during Christmas and other holiday meals.

...

Figure 13: Qualitative examples of the PPO models with different rewards. Different from both UltraRM and LI (Length Incentive only) rewards incur repetitions, RER based model provides more relevant and reliable responses.

Prompt: Provide me with a list of 10 names from various cultures that mean love, care, and/or empathy.	
RER: 1. Aahana (Hindi) 2. Ahava (Hebrew) 3. Ai (Japanese) 4. Agape (Greek) 5. Alay (Tagalog) 6. Betlehem (Hebrew) 7. Cariño (Spanish) 8. Daba (Amharic) 9. Dhul (Arabic) 10. Empatia (Polish)	LI · RP · QR: 1. Arabic: Rahmah 2. Chinese: Ai 3. French: Amour 4. Greek: Agape 5. Hindi: Sharada 6. Japanese: Ai ... 38. Zulu: Ngiyabonga 39. Vietnamese: Yêu Mây 40. Irish: Cead Mille Failte 41. Italian: Carità
Prompt: Write down three antonyms for the given word. 'laureating'	
RER: denigrating, disparaging, belittling.	LI · RP · QR: The antonyms of laureating are: - Decorating - Dignifying - Honoring - Magnifying - Praising - Sanctifying - Serving - Exalting - Glorifying ...

Figure 14: Qualitative examples comparing RER and LI · RP · QR against to CE type queries. LI · RP · QR often violates the given constraints, RER follows the given instruction much better.