RMBOOST: REWARD MODEL TRAINING WITH PREFERENCE-CONDITIONAL MULTI-ASPECT SYNTHETIC DATA GENERATION

Jiaming Shen [†] **Ran Xu** [‡] **Yennie Jun** [†] **Zhen Qin** [†] **Tianqi Liu** [†]

Carl Yang [‡*] **Yi Liang** [†] **Simon Baumgartner** [†] **Michael Bendersky** [†]

Google[†], Emory University[‡]
{jmshen, zhenqin, tianqiliu, yiliang, simonba, bemike}@google.com
{ran.xu, j.carlyang}@emory.edu

ABSTRACT

Reward models (RMs) are crucial for aligning large language models (LLMs) with human preferences. They are trained using preference datasets where each example consists of one input prompt, two responses, and a preference label. As curating a high-quality human labeled preference dataset is both time-consuming and expensive, people often rely on existing powerful LLMs for preference label generation. This can potentially introduce noise and impede RM training. In this work, we present RMBoost, a novel synthetic preference data generation paradigm to boost reward model quality. Unlike traditional methods, which generate two responses before obtaining the preference label, RMBoost first generates one response and selects a preference label, followed by generating the second more (or less) preferred response conditioned on the pre-selected preference label and the first response. This approach offers two main advantages. First, RMBoost reduces labeling noise since preference pairs are constructed intentionally. Second, RMBoost facilitates the creation of more diverse responses by incorporating various quality aspects (e.g., helpfulness, relevance, completeness) into the prompts. We conduct extensive experiments across three diverse datasets and demonstrate that RMBoost outperforms other synthetic preference data generation techniques and significantly boosts the performance of four distinct reward models.

1 INTRODUCTION

Large language models (LLMs) (Anil et al., 2023; OpenAI, 2023; Team et al., 2023; Touvron et al., 2023) have recently demonstrated unprecedented capabilities in various tasks. Leveraging a reward model (RM) to align LLMs with human preference (either through reinforcement learning (Ouyang et al., 2022; Stiennon et al., 2020) or via direct optimization over RM offline labeled preference pairs (Liu et al., 2024b; Rafailov et al., 2023; Yuan et al., 2024a)) is widely considered as a major breakthrough in modern LLM developments, when traditional supervised fine-tuning (SFT) alone yields suboptimal generation quality (Kirk et al., 2024).

To develop a high-quality reward model, it is necessary to collect a preference dataset comprising triplets of input prompt x, dual responses (y_1, y_2), and a response preference label l. As manually curating such preference dataset at scale is expensive, researchers have investigated automated methods for generating preference labels. One pioneering work, RLAIF (Bai et al., 2022), proposes to generate synthetic preference labels by prompting an LLM with few-shot side-by-side demonstrations (see Fig. 1(a)). Follow up studies (Lee et al., 2023; Pace et al., 2024) extend this idea by first distilling LLM few-shot predictions into an initial RM and then leveraging it to score candidate response pairs for selecting the preferred response. However, since the few-shot LLM predictor is not flawless, all

[*]Work done while being a visiting faculty in Google.

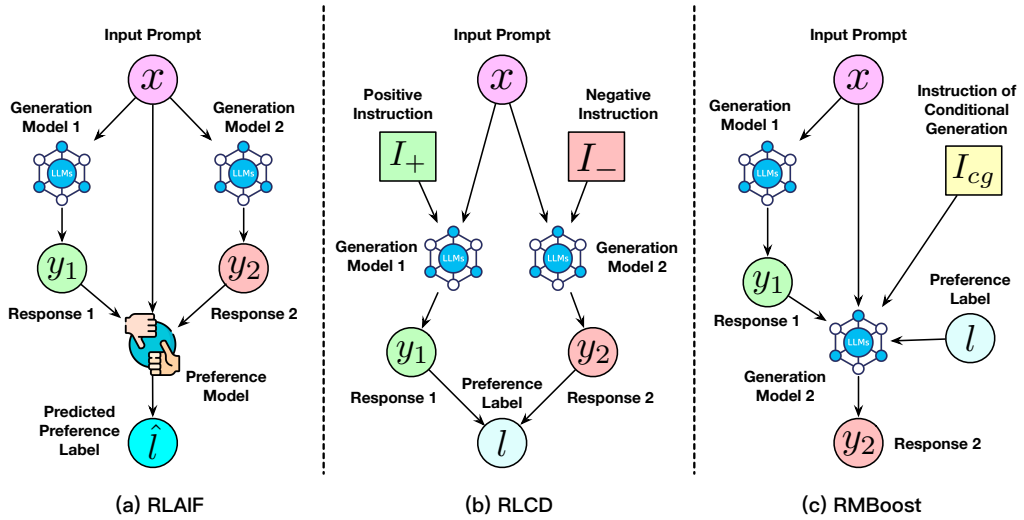

Figure 1: The illustration compares `RMBoost` with existing paradigms for synthetic preference data generation. The two generation models can correspond to the same model (configured at non-zero temperature) that samples two different responses. **(a)** The RLAIF approach first generates two responses and then leverages an initial preference model (e.g., LLM with few-shot side-by-side demonstrations) to predict the preference label. **(b)** The RLCD method produces two responses using contrasting prompts and assigns the preference label based on the respective prompt. **(c)** Our approach uniquely generates the second response conditioned on the first response and a predetermined preference label.

these methods produce noisier pairwise preferences than human-generated ones. In an alternative approach, RLCD (Yang et al., 2023) generates two responses with two contrasting prompts and obtains the preference label based on the prompt employed (see Fig. 1(b)). While this method reduces the preference labeling errors, these contrasting prompts typically focus on a single evaluation aspect (e.g., helpfulness), which restricts the diversity of the responses generated.

In this work, we present `RMBoost`, a novel synthetic preference data generation paradigm designed to boost the quality of reward models (see Fig. 1(c)). Our key innovations lie in the *progressive* way of generating preference pairs. Instead of predicting the preference label l for a pair of existing responses (y_1, y_2), `RMBoost` first generates one response y_1 and selects a preference label l. Then, `RMBoost` generates a second more (or less) preferred response y_2, conditioned on y_1 and l, and guided by predefined evaluation aspects (e.g., helpfulness, relevance, faithfulness, etc). In other words, `RMBoost` explicitly improves (or corrupts) the first response y_1 and transforms it into the second response y_2, thereby reducing preference label noise. Meanwhile, `RMBoost` leverages multi-aspect prompting to ensure that y_2 is sufficiently distinct from y_1, which not only provides fine-grained control over the generated text, but also helps to promote the diversity of generated datasets.

Intuitively, `RMBoost` improves performance over previous approaches based on two key observations: (1) existing methods encounter preference prediction errors when the model needs to weigh multiple evaluation aspects (Hong et al., 2023; Knox et al., 2024), and (2) LLMs exhibit strong conditional generation capabilities when provided with specific instructions (Ouyang et al., 2022). For instance, instructing an LLM to either corrupt or improve a response (i.e., y_1) with respect to one or more aspects typically yields highly effective results. Furthermore, `RMBoost` benefits from a balance between response distribution shift and label noise. Previous methods, which sample the second response y_2 from the same distribution as the LLM at inference time, tend to introduce more noise in the preference label prediction stage. In contrast, `RMBoost` samples y_2 from a modified distribution, which leads to a distribution shift but also reduces the preference prediction errors. In situations where the distribution shift is minor compared to the benefits of fewer preference errors, our method can significantly improve RM training.

We conduct extensive experiments between `RMBoost` and other leading synthetic preference data generation methods on three diverse datasets: QA Feedback (Wu et al., 2023), Ultra Feedback (Cui et al., 2023), and TLDR summarization (Stiennon et al., 2020). As detailed in §5, when generating

the preference data with PaLM 2-L (Anil et al., 2023) and GPT-4 (OpenAI, 2023), RMBoost substantially outperforms the established baselines in terms of preference prediction accuracy across four different RM backbones. Besides, our predictive gains can be successfully propagated to the alignment task: the LLM trained with our RMBoost *consistently* achieves higher win rate over baselines. Our analysis further confirms the benefit of RMBoost for improving the diversity of the responses and mimicing the style of ground-truth preference pairs.

2 Related Work

Synthetic Preference Data Generation. As reward models play a vital role in LLM developments (Ouyang et al., 2022; Touvron et al., 2023) and it is expensive to collect human preference data for RM training, a few works have leveraged LLMs for synthetic preference data generation. One early attempt, reinforcement learning from AI feedback (RLAIF) (Bai et al., 2022; Lee et al., 2023), prompts LLMs to rate preference labels for a given response pair. Later, the West-of-N (Pace et al., 2024) technique proposes to further bootstrap an existing RM by directly selecting the best and worst candidates in a pool of responses to a given query as the preference pairs. Along another line of work, ALMoST (Kim et al., 2023) proposes to query two LLMs of different qualities and assumes the response from a stronger model is preferred over the response from a weaker model. More recently, RLCD (Yang et al., 2023) improves RLAIF and ALMoST by adopting two contrasting prompts (one positive, one negative) to generate two responses and directly obtaining preference labels based on the prompts used. These works all demonstrate that LLMs can generate useful preference data for training reward models. At a high level, RMBoost is more related to RLCD, as both skip the preference label prediction step. RLCD achieves this implicitly via the contrasting prompts along one considered aspect (e.g., helpfulness or harmlessness). Our method, on the other hand, accomplishes this explicitly by feeding the preference labels directly into the prompt and enabling the LLMs to edit one response along multiple aspects. See Appendix A for a more extended review of related studies.

Attribute-aware Text Generation. Our work is also related to aspect/attribute-controlled text generation. One pioneering work (Logeswaran et al., 2018) shows that we can modify the style of a sentence while preserving its content using a small neural generation model. Follow up studies (Russo et al., 2020; Yu et al., 2021) extend this idea to sentiment and topic controlled text generation. Based on these findings, a more recent study (Yu et al., 2023a) proposes AttrPrompt, which leverages LLMs to generate synthetic data for classification tasks. Our method is related to AttrPrompt in the sense that we all aim to increase the diversity of generated text by leveraging multi-aspect controlled generation. However, we differ significantly in terms of the targeted aspects, downstream tasks and input formats (i.e., sentiment/topic for single-sentence classification tasks in AttrPrompt versus helpfulness/relevance for RM training in RMBoost).

3 Preliminaries

In this work, we use $\mathcal{X}$ and $\mathcal{Y}$ to denote the space of model input prompts and model output responses. Furthermore, we denote the language model to be aligned with human preference as $\pi : \mathcal{X} \rightarrow \mathcal{Y}$ and represent the pointwise reward model as $r : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$.

Preference Data and Reward Modeling. The reward model r is typically trained on a preference dataset $\mathcal{D}_{\text{RM}} = \{(x^i, y_1^i, y_2^i, l^i)\}|_{i=1}^N$ where each example consists of an input prompt x^i, two responses y_1^i, y_2^i, and a preference label $l^i \in \{-1, 1\}$ indicating which response is preferred. We denote the preferred response as y_+^i and the less preferred one as y_-^i.

Following the Bradley-Terry (Bradley & Terry, 1952) assumption, we train the reward model by minimizing the following empirical negative log-likelihood loss:

$$L(r_\theta, \mathcal{D}_{\text{RM}}) = -\mathbf{E}_{(x,y_+,y_-)\in\mathcal{D}_{\text{RM}}}[\log(\sigma(r_\theta(x, y_+) - r_\theta(x, y_-)))], \tag{1}$$

where r_θ is the reward model parameterized by θ and σ denotes the sigmoid function.

Synthetic Preference Data Generation. The pioneering work (Stiennon et al., 2020) constructs the preference dataset entirely through manual curation, aptly named "Reinforcement Learning from Human Feedback". Later studies (Bai et al., 2022; Lee et al., 2023) propose to replace the human

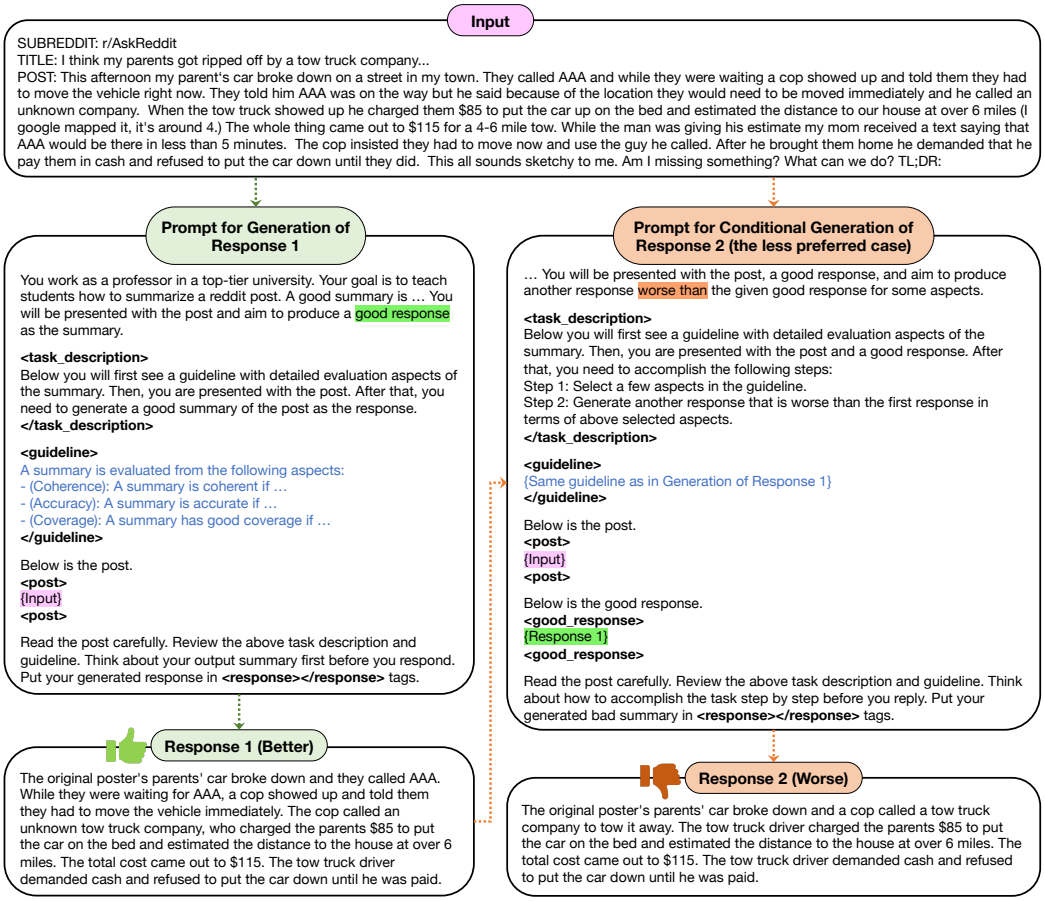

Figure 2: Overview of our `RMBoost` framework with prompt gists. See §H for the complete prompts.

feedback with LLM generated preference labels (in other words, AI Feedback). Specifically, they first assume the access to a set of unlabeled prompts $\{x \in \mathcal{X}\}$. Then, for each input prompt x^i, they generate two responses y_1^i, y_2^i by sampling from either the same LLM twice or from two different LLMs. After that, another strong off-the-shelf LLM (e.g., GPT4, Gemini Ultra) is leveraged to generate the synthetic preference label $\hat{l^i}$ (which could be different from the real preference label l^i). Finally, by repeating this process for all input prompts, we obtain a synthetic preference dataset $\mathcal{D}_{\text{SYN}} = \{(x^i, y_1^i, y_2^i, \hat{l^i})\}|_{i=1}^N$ for training reward models.

4 The RMBoost Framework

We first describe our method, `RMBoost`, a novel method for boosting reward model training with preference-conditional multi-aspect synthetic data generation. Then, we present its high-level intuitions, followed by a more theoretical analysis.

4.1 Method Description

`RMBoost` begins with a collection of unlabeled input prompts $\{x \in \mathcal{X}\}$, similar to RLAIF or RLCD, and utilizes an off-the-shelf LLM π for generating responses. For each input prompt x^i, our method initially samples one response y_1^i from π. Subsequently, a preference label l^i (designated as "more preferred" or "less preferred") is predetermined. The LLM is then prompted to generate a second response y_2^i, this time conditioned on the first response y_1^i, the preference label l^i, and a specific instruction for conditional generation I_{cg}. This instruction outlines all the relevant aspects for evaluating the response, such as helpfulness, relevance, and coherence, and guides the LLMs to adjust the first response according to these criteria. Then, we form the preference pair (y_+^i, y_-^i) to be (y_1^i, y_2^i)

if the preference label l^i is designated as "less preferred", or (y_2^i, y_1^i) if l^i is "more preferred". At last, we can optionally use the standard RLAIF method to check the quality of constructed preference pairs and filter those low quality ones (See Appendix B for more discussions).

Figure 2 shows one example where we select the preference label l as "less preferred" and intentionally let the LLM corrupt the first tl;dr summary to become worse along the "coverage" and "accuracy" aspects. After constructing the synthetic preference data, `RMBoost` follows the standard practice to train the reward model. The learned reward model can then be employed for LLM alignment.

4.2 Intuitions for RMBoost

Intuitively, `RMBoost` surpasses existing methods based on two key observations: (1) many preference prediction errors in previous methods occur when the model must consider multiple evaluation aspects simultaneously, and (2) the LLM demonstrates robust directional generation capabilities when given explicit instructions. Specifically, when instructed to modify a response (i.e., y_1) with respect to one or more aspects, the LLM typically excels.

Another perspective on the effectiveness of `RMBoost` is its ability to balance between response distribution shifts and label noise. Traditional methods sample the second response y_2 from the same distribution $\Pr(y|x)$ as the LLM's inference time distribution. However, they need another step to prompt the LLM for preference prediction, which often introduces many labeling noise, especially when comparing two responses requires thorough considerations over multiple evaluation aspects (i.e., our first key observation). In contrast, `RMBoost` samples y_2 from a modified distribution $\Pr(y|x, y_1, l, I_{cg})$, resulting in a distribution shift but experiencing fewer preference prediction errors (due to our second key observation). If this distribution shift is relatively minor compared to the reduction in preference prediction errors, our method can significantly enhance RM training. We present this analysis more formally below.

4.3 Analysis of RMBoost and Comparisons with Previous Approaches

Reward model training on a clean human-labeled preference dataset can be viewed as minimizing an empirical version of the following risk function:

$$\mathbf{E}_{(x,y_1,y_2,l)\sim\Pr^*}[L(x, y_1, y_2, l; \theta)], \tag{2}$$

where l is the *ground truth* preference label, L is the loss function, θ is the RM parameter set, and $\Pr^*$ denotes the true (human-labeled) data distribution, defined as follows[1]:

$$\Pr^*(x, y_1, y_2, l) = \Pr(x)\Pr(y_1|x)\Pr(y_2|x)\Pr^*(l|x, y_1, y_2). \tag{3}$$

This joint distribution reflects the common human and AI preference data collection processes.

RLAIF. When the preference labels are machine generated, we use $\tilde{l}$ to denote the noisy preference label and follow previous literature (Sukhbaatar et al., 2015) to assume a noise corruption distribution $Q(\tilde{l}|l)$. Then, we can define the risk on this "noisy" preference dataset as follows:

$$\mathbf{E}_{(x,y_1,y_2,\tilde{l})\sim\Pr^{\text{RLAIF}}}[L(x, y_1, y_2, \tilde{l}; \theta)], \tag{4}$$

where the noisy data distribution (for RLAIF approaches) is:

$$\Pr^{\text{RLAIF}}(x, y_1, y_2, \tilde{l}) = \sum_l \Pr(x)\Pr(y_1|x)\Pr(y_2|x)\Pr^*(l|x, y_1, y_2)Q(\tilde{l}|l). \tag{5}$$

By plugging in Eq. 5 into the above Eq. 4, we have:

$$\mathbf{E}_{(x,y_1,y_2,\tilde{l})\sim\Pr^{\text{RLAIF}}}[L(x, y_1, y_2, \tilde{l}; \theta)] = \mathbf{E}_{(x,y_1,y_2,\tilde{l})\sim\Pr^*}\left[\beta^{\text{RLAIF}}(x, y_1, y_2, \tilde{l})L(x, y_1, y_2, \tilde{l}; \theta)\right] \tag{6}$$

$$\beta^{\text{RLAIF}}(x, y_1, y_2, \tilde{l}) = \frac{\Pr^{\text{RLAIF}}(x, y_1, y_2, \tilde{l})}{\Pr^*(x, y_1, y_2, \tilde{l})}. \tag{7}$$

[1]Without the loss of generality, we assume two responses are sampled from the same LLM independently. If they come from two different LLMs, we can replace the second $\Pr(y_2|x)$ with another distribution and the below derivations should still hold.

The left hand side of Eq. 6 is the raw training objective. The right hand side of Eq. 6 is essentially a sample re-weighted version of above Eq. 2. In other words, when we consider the preference label noise, we are optimizing a biased risk function on clean data. Furthermore, the closer the re-weighting factor $\beta^{\text{RLAIF}}(x, y_1, y_2, \tilde{l})$ is to 1, the less bias we have on the training objective of existing methods.

Let dive into the re-weighting factor by combining Eq. 5 with Eq. 7, we have:

$$\beta^{\text{RLAIF}}(x, y_1, y_2, \tilde{l}) = \frac{\sum_l \Pr^*(l|x, y_1, y_2) Q(\tilde{l}|l)}{\Pr^*(\tilde{l}|x, y_1, y_2)} = Q(\tilde{l}|\tilde{l}) + \frac{\sum_{l \neq \tilde{l}} \Pr^*(l|x, y_1, y_2) Q(\tilde{l}|l)}{\Pr^*(\tilde{l}|x, y_1, y_2)}. \tag{8}$$

If we consider a simple binary preference prediction setting where either y_1 or y_2 is preferred without tie. Given a noisy example where the observed label $\tilde{l}$ is different from its true label l, the second term in the right-hand side of Eq. 8 will explode to a very large number. This is because its denominator $\Pr^*(\tilde{l}|x, y_1, y_2)$, the ground truth probability of getting a wrong prediction label $\tilde{l}$, is close to 0, while the nominator $\Pr^*(l|x, y_1, y_2)$ is close to 1. Consequently, $\beta^{\text{RLAIF}}(x, y_1, y_2, \tilde{l})$ is far away from 1 and we are optimizing a very biased version of the true risk function.

RLCD. The synthetic preference pairs generated by RLCD follow the below distribution[2]:

$$\Pr^{\text{RLCD}}(x, y_1, y_2, l) = \Pr(x) \Pr(l) \Pr(y_1|x, I_+) \Pr(y_2|x, I_-), \tag{9}$$

where I_+ (I_-) denotes the prompt for generating the positive (negative) response, respectively. Following the same derivations above, we will have:

$$\beta^{\text{RLCD}}(x, y_1, y_2, l) = \frac{\Pr(l) \Pr(y_1|x, I_+) \Pr(y_2|x, I_-)}{\Pr(y_1|x) \Pr(y_2|x) \Pr^*(l|x, y_1, y_2)}, \tag{10}$$

Here, we first notice that $\Pr(l)$ and $\Pr^*(l|x, y_1, y_2)$ are essentially preference label frequency distribution and they won't matter if we adopt the example flipping strategy[3]. The remaining terms indicate the distribution shift between real responses and synthetically generated responses.

RMBoost. In our synthetic data generation method, we directly sample the true preference label l and generate the second response y_2. Therefore, the distribution of our synthetic data is:

$$\Pr^{\text{RMBoost}}(x, y_1, y_2, l) = \Pr(x) \Pr(y_1|x) \Pr(l) \Pr(y_2|x, y_1, l, I_{cg}), \tag{11}$$

where I_{cg} is a fixed conditional generation instruction. We then follow the same derivation of Eq. 6 and obtain our re-weighting factor as follows:

$$\beta^{\text{RMBoost}}(x, y_1, y_2, l) = \frac{\Pr(l) \Pr(y_2|x, y_1, l, I_{cg})}{\Pr^*(l|x, y_1, y_2) \Pr(y_2|x)}. \tag{12}$$

Similar to the above RLCD derivation, we can cancel out the terms $\Pr(l)$ and $\Pr^*(l|x, y_1, y_2)$. The remaining terms $\Pr(y_2|x, y_1, l, I_{cg})$, $\Pr(y_2|x)$ correspond to the second response distributions. If these two distributions are close to each other, we will have a $\beta^{\text{RMBoost}}(x, y_1, y_2, l)$ closer to 1, which enables us to optimize a less biased version of the true risk function. In the below experiment §5.5, we empirically show that `RMBoost` indeed produces a distribution of re-weighting factor β closer to 1 and facilitates the reward model training with a less biased objective.

5 Experiments

5.1 Experiment Settings

Datasets and Tasks. We analyze and evaluate all compared methods on three diverse sets of datasets. (1) **QA Feedback** (Wu et al., 2023) is a long-form QA dataset where the model inputs a question, a set of Wikipedia passages and outputs a long-form response to answer the given question. The raw dataset contains both supervised finetune (SFT) data and reward model (RM) training data. For SFT

[2]We assume the first response is more preferred. Due to symmetries, below derivations hold for the case where the second response is more preferred.

[3]The example flipping strategy means we can flip y_1, y_2 along with their corresponding preference label and construct a "new" reward model training example, which effectively makes both $\Pr(l)$ and $\Pr^*(l|x, y_1, y_2)$ to be uniform distribution and cancel out.

Table 1: Statistics of datasets.

Dataset	Task	# SFT Train	# SFT Val	# RM Train	# RM Val	# RM Test
QA Feedback (Wu et al., 2023)	Question Answering	1,000	500	14,982	1,344	1,272
Ultra Feedback (Cui et al., 2023)	General LLM Alignment	13,920	8,477	33,897	4,238	4,239
TLDR Summarization (Stiennon et al., 2020)	Summarization	116,722	6,447	92,534	83,797	83,629

data, we re-use its original train/val splits. For RM data, we construct preference pairs by considering all pairwise responses in the raw data and split its original validation set into new val/test sets. (2) **Ultra Feedback** (Cui et al., 2023) is a large-scale, diversified preference dataset built for general LLM alignment research. Each example in this benchmark includes one prompt and 4 responses associated with their quality scores. As the original benchmark does not contain SFT data and its preference data have no standard train/val/test splits, we create our own as follows. For each example, we first select the response with the best overall score as the candidate preferred response, and the one with the least overall score as the candidate not preferred response. Then, we check if the score of the candidate preferred response is larger than the score of the candidate not preferred response by at least 1.5. If yes, we construct a preference pair. Otherwise, we place this candidate preferred response along with its prompt into the SFT data. (3) **TLDR Summarization** (Stiennon et al., 2020) consists of Reddit posts along with their human written summaries (for SFT) and pairs of machine-generated summaries rated by human labels (for RM training). We use the existing splits in the original dataset.

Compared Methods. We compare our `RMBoost` method with the following synthetic preference data generation baselines: (1) **RLAIF** (Bai et al., 2022), which uses LLMs to rate preference labels; (2) **West-of-N** (Pace et al., 2024), which directly selects the best and worst candidates in a pool of responses to a given query; (3) **RLCD** (Yang et al., 2023), which leverages two contrasting prompts to generate two responses and returns the response associated with the positive prompt as the preference response. For each compared method as well as `RMBoost`, we report their associated prompt templates in Appendix H.

Backbone LLMs. For synthetic preference data generation, we use PaLM 2-L (Anil et al., 2023) in our main experiments. We also evaluate `RMBoost` with GPT-3.5 (OpenAI, 2022) and GPT-4 (OpenAI, 2023) in §5.4. For RM training (on both real and synthetic preference data as well as their mixture), we investigate four backbone models: Gemini-Nano-1, Gemini-Nano-2 (Team et al., 2023), PaLM 2-XXS (Anil et al., 2023), and Gemma 2B (Mesnard et al., 2024).

Evaluation Protocols. We first train the model on each dataset's SFT data and select the checkpoint with the highest performance metric on the SFT validation set. Then, we initialize the reward model with the above selected SFT checkpoint and continue fine-tune it on the reward model training set. We use each dataset's RM validation set to select hyper-parameters and report the preference prediction accuracy of each fine-tuned RM on the RM test set. Appendix C includes more details.

5.2 Main Experiment Results

In our main experiments, we first use different synthetic data generation methods to generate preference datasets. Then, we train various backbone reward models on three dataset settings: (1) **Real**: only human rated dataset, (2) **Syn**: only synthetically generated dataset, and (3) **Real + Syn**: mixture of both human rated and synthetically generated datasets. Results are shown in Table 2, from which we have the following findings. We observe that mixing synthetic data with human labeled ones in general can help the reward model training. Among all the methods, `RMBoost` generally outperforms other baseline methods in both synthetic and mixtures variants, indicating the high quality of generated synthetic preference data. Besides, when mixing output synthetic data with the real data, `RMBoost` gives the most performance boost, which highlights that `RMBoost` effectively complements the ground-truth data to strengthen reward modeling.

5.3 Ablations of RMBoost

We continue to evaluate a few variants of `RMBoost` and study how different design choices affect its performance. First, we compare `RMBoost` with its **"No-Aspect"** version where we intentionally remove the detailed aspect definitions in response generation instruction and replace them with general words like "good/bad response". Second, we test a **"No-Filtering"** variant which skips the

Table 2: Overall experiment results with PaLM 2-L as the preference data generation model. We train each reward model backbone on preference data generated by each method. Specifically, "Real" ("Syn") indicates that RM is trained only on the real (synthetic) preference data and "Real + Syn" means the RM is fine-tuned on a mixture of real and synthetic preference data. All numbers are the preference prediction accuracy of a fine-tuned RM on each dataset's test set.

Backbone	**Methods**	**QA Feedback**			**Ultra Feedback**			**TLDR Summarization**		
		Real	Syn	Real + Syn	Real	Syn	Real + Syn	Real	Syn	Real + Syn
Gemini-Nano-1	RLAIF	63.67	56.25	61.33	79.69	62.89	76.17	74.61	68.75	73.83
	West-of-N		59.38	60.94		73.05	88.28		70.31	75.00
	RLCD		60.16	64.06		73.44	88.67		70.70	76.56
	RMBoost		**61.33**	**67.97**		**74.61**	**90.23**		**71.48**	**77.34**
Gemini-Nano-2	RLAIF	67.58	59.38	65.23	92.97	75.00	89.06	80.47	71.48	79.69
	West-of-N		60.55	65.63		76.56	90.63		71.88	80.08
	RLCD		**61.33**	66.80		**79.69**	93.75		**73.83**	81.26
	RMBoost		60.55	**68.36**		79.30	**94.14**		73.44	**82.81**
PaLM 2-XXS	RLAIF	70.31	57.81	64.06	90.63	75.26	89.56	71.48	63.28	70.31
	West-of-N		59.38	67.17		76.56	92.97		64.45	71.09
	RLCD		60.94	70.31		**78.13**	**93.75**		**67.67**	72.27
	RMBoost		**64.06**	**75.00**		**78.13**	**93.75**		66.80	**72.66**
Gemma 2B	RLAIF	60.05	51.17	56.64	86.28	60.94	82.81	70.69	62.50	67.67
	West-of-N		54.30	57.03		65.63	85.94		63.28	70.70
	RLCD		55.86	60.55		68.65	87.50		64.45	71.48
	RMBoost		**56.92**	**61.47**		**68.71**	**87.88**		**65.65**	**71.53**

Table 3: Ablations of RMBoost with Gemini-Nano-1 as the backbone RM.

Method	**QA Feedback**	**Ultra Feedback**	**TLDR Summarization**
RMBoost	67.97	**90.23**	77.34
No-Aspect	63.67	88.28	75.78
No-Filtering	67.41	89.06	77.34
+SFT-Response	**68.75**	89.45	**79.30**

Table 4: RMBoost with GPT-3.5 and GPT-4 as synthetic data generation (DGen) models.

RM Backbone	**DGen Model**	**QA Feedback**			**TLDR Summarization**		
		Real	Syn	Real + Syn	Real	Syn	Real + Syn
Gemini-Nano-1	GPT-3.5	**63.67**	58.59	63.28	74.61	71.88	**77.73**
	GPT-4	63.67	60.16	**66.02**	74.61	73.05	**78.52**
Gemini-Nano-2	GPT-3.5	67.58	60.94	**67.91**	80.47	72.27	**81.97**
	GPT-4	67.58	64.06	**68.75**	80.47	73.83	**83.26**

post generation quality check step (see §4.1) and directly use all generated preference pairs to train reward models. We run these ablation studies with Gemini-Nano-1 as the backbone model and report the results in Table 3. We can see that both variants perform worse than the full RMBoost by varying degrees. Specifically, we observe that removing the multiple aspect definitions will significantly hurt the quality of generated data (and thus the RM quality). Meanwhile, skipping the post-generation quality check step has smaller (though negative) effects can could be considered as an option when the computation resources are limited (for LLM bulk inference).

Additionally, we test a variant of RMBoost that directly leverages the SFT response as the "preferred" response and generates a less preferred one as its counterpart. We denote this variant as "**+SFT-Response**". From Table 3, we can see that this variant can outperform the vanilla RMBoost on two out of three datasets. Although this improvement is somewhat expected as additional high-quality signals from the SFT data are introduced, we think this could be an interesting variant for two reasons. First, it allows us to skip the first response generation step and thus saves the computes. Second, it enables people to construct a preference dataset from an SFT dataset by utilizing the conditional generation module in RMBoost.

5.4 Experiments with More Synthetic Data Generation Models

We showcase the versatility of RMBoost by employing GPT-3.5 (OpenAI, 2022) and GPT-4 (OpenAI, 2023) as data generation models. Specifically, we utilize Gemini-Nano-1 and Gemini-Nano-2 as the backbone reward models, training them on the datasets generated above. Table 4 presents the results on two datasets. From the results, we observe that employing a more powerful LLM as the backbone enhances the quality of synthetic preference data, aligning with the intuition that larger models excel in instruction comprehension. Furthermore, we note that RMBoost serves as a versatile data generation pipeline, adaptable to different generation models, thus producing high-quality data to enhance reward model performance across diverse datasets.

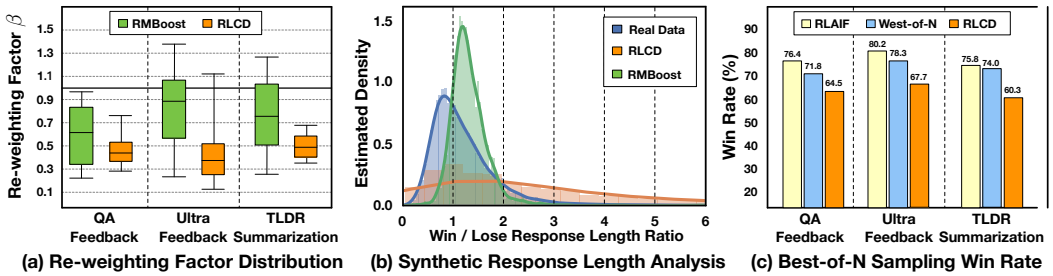

Figure 3: **(a)** Distributions of response re-weighting factor β. **(b)** Histograms of win and lose response length ratio. **(c)** The win rates of `RMBoost` over compared baselines for aligning LLMs.

5.5 Analysis of Generated Synthetic Data

Below, we analyze the quality of our synthetically generated datasets from two aspects. First, as discussed in §4.3, RM training on the synthetic preference data is equivalent to minimizing a biased risk function. The bias is determined by a per-sample re-weighting factor β. Therefore, we first analyze the distributions of this factor β by evaluating `RMBoost` and RLCD (the best performing baseline) with Gemini-Nano-1 as the backebone RM. Results are plotted in Figure 3(a). We can see that `RMBoost` overall produces synthetic preference datasets that are more similar to the real datasets (the median β is closer to 1). Furthermore, the responses generated from `RMBoost` have a larger range of β value, which indicates that they are more diverse.

Second, we analyze the length of generated preference pairs. Specifically, we compute the length ratio of a preferred (win) response over a less preferred (lose) response. Figure 3(b) plots a histogram of these length ratios on the QA Feedback dataset. We observe that both our method and RLCD generally prefer longer responses than shorter ones (as the modes of both methods are larger than 1). In addition, we can see that the length ratio distribution of `RMBoost` generations is closer to the real data, which partially explains the success of our approach.

5.6 Cost and Efficiency of RMBoost

To label each synthetic preference example, `RMBoost` will call the LLM twice for generating two responses, similar to RLCD. However, both the input prompt (which includes the first generated response) and the output response of `RMBoost` is longer than RLCD, which leads to longer decoding time and larger cost. Empirically, we observe that the total inference cost of `RMBoost`, measured in the number of tokens, is about 30% larger than RLCD. This overhead can be potentially reduced when we leverage a pre-existing response (e.g., the SFT response).

5.7 Measuring RM Quality for LLM Alignment

We continue to evaluate the quality of these RMs for aligning LLMs. For each dataset, we employ the *best-of-N* sampling by first leveraging the SFT checkpoint to generate 9 responses, and then selecting the one with the highest reward score from each trained RM. Then, we train a large Gemini-Pro model on each dataset's original RM training data and utilize it as the side-by-side auto-rater. Finally, we compare the responses selected by `RMBoost` with those selected by each individual RM using this auto-rater and report the results in Figure 3(c). We observe that the RM trained on `RMBoost` can generally select better responses than RMs trained on other synthetic data generation methods, and such gains can be further propagated to downstream alignment tasks. This demonstrates that `RMBoost` enables developer to obtain a better reward model for aligning LLMs.

6 Limitations and Impact & Ethics Statement

One limitation of `RMBoost` is that it requires many computational resources for LLM bulk inferences. In addition, `RMBoost` assumes that prompt engineers are fully aware of all evaluation criteria when crafting multi-aspect generation instructions. Finally, like many other synthetic data generation methods, `RMBoost` could potentially generate plausible but inaccurate information. Overall, `RMBoost`

has a positive social impact by enabling developers to improve reward models, thereby better aligning LLMs with human values. However, there is a risk that RMBoost could be misused to introduce biases or harmful contents into synthetic datasets intentionally.

7 Conclusions and Future Work

This study introduces RMBoost, an innovative approach to generating synthetic preference data for training RMs. By systematically generating responses conditioned on pre-selected preference labels, RMBoost effectively minimizes the label noise, a common limitation in previous methods. Our experiments validate that RMBoost enhances the accuracy of RMs across various datasets, establishing a new approach for synthetic preference data generation. Looking ahead, future research could further optimize RMBoost by exploring additional dimensions of response generation and refinement. Additionally, expanding RMBoost to domains with multi-model data inputs could be interesting direction. Ultimately, enhancing the robustness and scalability of RMBoost will be crucial for its adoption in broader LLM applications, making this an exciting direction for subsequent advancements in the field.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, John Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, E Perez, Jamie Kerr, Jared Mueller, Jeff Ladish, J Landau, Kamal Ndousse, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noem'i Mercado, Nova Dassarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Sam Bowman, Zac Hatfield-Dodds, Benjamin Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom B. Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback. *ArXiv*, abs/2212.08073, 2022. URL https://api.semanticscholar.org/CorpusID:254823489.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs the method of paired comparisons. *Biometrika*, 39:324–345, 1952. URL https://api.semanticscholar.org/CorpusID:121987403.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *ArXiv*, abs/2310.01377, 2023. URL https://api.semanticscholar.org/CorpusID:263605623.

Qingxiu Dong, Li Dong, Xingxing Zhang, Zhifang Sui, and Furu Wei. Self-boosting large language models with synthetic preference data. *arXiv preprint arXiv:2410.06961*, 2024.

Joey Hong, Kush Bhatia, and Anca Dragan. On the sensitivity of reward inference to misspecified human models. In *ICLR*, 2023. URL https://openreview.net/forum?id=hJqGbUpDGV.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In *ACL*, 2023.

Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Min Yoo, and Minjoon Seo. Aligning large language models through synthetic feedback. *EMNLP*, abs/2305.13735, 2023. URL https://api.semanticscholar.org/CorpusID:258841835.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *ICLR*, 2024. URL https://openreview.net/forum?id=PXD3FAVHJT.

W. Bradley Knox, Stephane Hatgis-Kessell, Serena Booth, Scott Niekum, Peter Stone, and Alessandro G Allievi. Models of human preference for learning reward functions. *TMLR*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=hpKJkVoThY.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *ArXiv*, abs/2309.00267, 2023. URL https://api.semanticscholar.org/CorpusID:261493811.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*, 2024.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. Best practices and lessons learned on synthetic data for language models. 2024a. URL https://api.semanticscholar.org/CorpusID:269042851.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. *ICLR*, 2024b.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. Content preserving text generation with attribute controls. *Advances in Neural Information Processing Systems*, 31, 2018.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. *NeurIPS*, 35:462–477, 2022.

Gemma Team Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, L. Sifre, Morgane Riviere, Mihir Kale, J Christopher Love, Pouya Dehghani Tafti, L'eonard Hussenot, et al. Gemma: Open models based on gemini research and technology. *ArXiv*, abs/2403.08295, 2024. URL https://api.semanticscholar.org/CorpusID:268379206.

OpenAI. Chatgpt, 2022.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *NeurIPS*, abs/2203.02155, 2022. URL https://api.semanticscholar.org/CorpusID:246426909.

Aliz'ee Pace, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. West-of-n: Synthetic preference generation for improved reward modeling. *ArXiv*, abs/2401.12086, 2024. URL https://api.semanticscholar.org/CorpusID:267069315.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2023.

Giuseppe Russo, Nora Hollenstein, Claudiu Musat, and Ce Zhang. Control, generate, augment: A scalable framework for multi-attribute text generation. *EMNLP Findings*, 2020.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. *ICML*, abs/2302.00618, 2023. URL https://api.semanticscholar.org/CorpusID:256459681.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir D. Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *ICLR*, 2015.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *NeurIPS*, 36, 2023.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, et al. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023. URL https://api.semanticscholar.org/CorpusID:259950998.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *ACL*, 2023. URL https://api.semanticscholar.org/CorpusID:254877310.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *NeurIPS*, abs/2306.01693, 2023. URL https://api.semanticscholar.org/CorpusID:259064099.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. Rlcd: Reinforcement learning from contrastive distillation for language model alignment. 2023. URL https://api.semanticscholar.org/CorpusID:260357852.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. Zerogen: Efficient zero-shot learning via dataset generation. In *EMNLP*, pp. 11653–11669, 2022.

Dian Yu, Zhou Yu, and Kenji Sagae. Attribute alignment: Controlling text generation from pre-trained language models. *EMNLP Findings*, 2021.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J. Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. Large language model as attributed training data generator: A tale of diversity and bias. *NeurIPS (Datasets and Benchmark Track)*, abs/2306.15895, 2023a. URL https://api.semanticscholar.org/CorpusID:259275123.

Yue Yu, Yuchen Zhuang, Rongzhi Zhang, Yu Meng, Jiaming Shen, and Chao Zhang. Regen: Zero-shot text classification via training data generation with progressive dense retrieval. *ACL Findings*, 2023b.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. *NeurIPS*, 36, 2024a.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=0NphYCmgua.

A More Related Work

Synthetic Preference Data Generation. LLMs have proven effective for serving as training data generators across diverse applications, but most works emphasize on task-specific finetuning (Meng et al., 2022; Shao et al., 2023; Ye et al., 2022; Yu et al., 2023b), instruction finetuning (Wang et al., 2023; Honovich et al., 2023; Peng et al., 2023; Xu et al., 2023; Li et al., 2024; Liu et al., 2024a) and alignment (Dong et al., 2024; Yuan et al., 2024b). As the reward model (RM) plays a vital role in LLM developments (Ouyang et al., 2022; Touvron et al., 2023) and it is expensive to collect human preference data for RM training, a few recent works have leveraged LLMs for synthetic preference data generation. One early attempt, known as reinforcement learning from AI feedback (RLAIF) (Bai et al., 2022; Lee et al., 2023), uses LLMs with few-shot demonstrations to rate preference labels for a given response pair. Later, the West-of-N (Pace et al., 2024) technique was developed to further bootstrap an existing RM by directly selecting the best and worst candidates in a pool of responses to a given query as the preference pairs. Along another line of work, ALMoST (Kim et al., 2023) proposes to query two LLMs of different qualities and assumes the response from a stronger model (e.g., GPT-4) is preferred over the response from a weaker model (e.g., LLaMA-7B). More recently, RLCD (Yang et al., 2023) improves RLAIF and ALMoST based on the context distillation idea (Sun et al., 2023). Specifically, RLCD adopts two contrasting prompts (one positive, one negative) to generate two responses and directly obtains preference labels based on the prompts used. These works all demonstrate that LLMs can generate useful preference data for training reward models. At a high level, `RMBoost` is more related to RLCD, as both skip the preference label prediction step. RLCD achieves this implicitly via the contrasting prompts along one considered aspect (e.g., helpfulness or harmlessness). Our method, on the other hand, accomplishes this explicitly by feeding the preference labels directly into the prompt and enabling the LLMs to edit one response along multiple aspects.

Attribute-aware Text Generation. Our study is also related to aspect/attribute-controlled text generation. One pioneering work (Logeswaran et al., 2018) shows that we can modify the style of a sentence while preserving its content using a small neural generation model. Follow up studies (Russo et al., 2020; Yu et al., 2021) extend this idea to sentiment and topic controlled text generation. Based on these findings, a more recent study (Yu et al., 2023a) proposes AttrPrompt, which leverages LLMs to generate synthetic data for classification tasks. Our method is related to AttrPrompt in the sense that we all aim to increase the diversity of generated text by leveraging multi-aspect controlled generation. However, we differ significantly in terms of the targeted aspects, downstream tasks and input formats (i.e., sentiment/topic for single-sentence classification tasks in AttrPrompt versus helpfulness/relevance for RM training in `RMBoost`).

B Optional Preference Pair Filtering Step Details

After constructing the preference pairs via `RMBoost`, we can optionally perform an additional quality check step. This step aims to reduce potential sampling errors from the conditional generation phase and to filter those low quality pairs. It is fully automated — we use the standard RLAIF method to verify the preference order of generated pairs, retaining only those that match the order established by `RMBoost`. In our experiments, we found the filtering rate is approximately 25% for `RMBoost`. We also added the same filtering step to RLCD and found that the filtering rate for RLCD is significantly lower (less than 5%), likely because RLCD modifies the response by only one aspect at a time.

C Implementation Details

For Gemini-Nano-1, Gemini-Nano-2, and PaLM 2-XXS are the RM backbones, we do parameter swamping on the learning rate in [3e-6, 1e-6, 3e-5, 1e-5, 3e-4, 1e-4], batch size in [8, 16, 32]. On both QA Feedback and Ultra Feedback datasets, the final selected hyper-parameters for Gemini-Nano-1

are learning rate = 1e-4, batch size = 16. For Gemini-Nano-2, learning rate = 3e-6, batch size = 16, and for PaLM 2-XXS, learning rate = 1e-5, batch size = 8. On the TLDR summarization dataset, the final selected hyper-parameters for Gemini-Nano-1 is learning rate = 1e-4, batch size = 32. For Gemini-Nano-2, learning rate = 3e-6, batch size = 16, and for PaLM 2-XXS, learning rate = 3e-4, batch size = 8. We train all these backbone RMs on in-house infrastructure with TPU v5.

For the open-source LLM model (Gemma 2B), we set the learning rate to 3e-6 for QAFeedback and UltraFeedback and 5e-6 for TL;DR dataset with batch size 64, and set the training step to 3 epochs. We use AdamW (Loshchilov & Hutter, 2017) as the optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ cosine scheduler and warmup for the first 3% steps. All the experiments are done using on 4 NVIDIA A100 40G GPUs.

D Experiments on Mistral-7B

To further demonstrate our method's generalization ability to the open-source model, we add one experiment with the Mistral-7B model. Following the same experiment protocol, we first do SFT on Mistral-7B-v0.1 to for RM initialization and then fine the RM using synthetically generated data. On the Ultra-feedback dataset, we observe a performance improvement from 87.43 to 87.9 and on the TLDR_summarization dataset, we improve the performance from 74.65 to 75.01. Both improvements are statistically significant (under paired samples t-test with p-value < 0.05).

E Fine-grained Performance Improvement

As `RMBoost` generates preference data along a set of evaluation aspects, we conduct one further experiment to check how these syntactic data improve the RM training in a more fine-grained way. Our experiments on the Ultra Feedback dataset, which provides per-aspect human ratings, show `RMBoost` most significantly improves the "Honesty" aspect (+1.67 rating), while the "Verbalized Calibration" aspect gets the least improvement (+0.12 rating). This observation also partially demonstrates the quality of `RMBoost` generated synthetic data, because otherwise, we will witness some performance drops on at least one aspect.

F Cost of RMBoost

Using OpenAI's bulk inference APIs, the QA-feedback dataset costs \$1.638 for GPT-3.5 and \$41.417 for GPT-4; the TLDR summarization dataset costs \$5.71 for GPT-3.5 and \$148.79 for GPT-4. These numbers are calculated based the OpenAI's pricing strategy back around April 2024. The costs for using PaLM 2-L (from public APIs) are on a similar scale to GPT-4. We believe these methods are generally more cost-effective than expert curations.

G Scaling Law of Synthetic Data

When designing our data mixture strategy, we initially aimed for a 1:1 ratio of synthetic to real data. In practice, however, we apply automated rules to filter out low-quality synthetic pairs. As a result, in the "Real+Syn" configuration, synthetic data typically comprises between 30-50% of the total dataset. A pilot study using the QA feedback dataset, where synthetic and real data were mixed at a 7:1 ratio, revealed that such an imbalance can actually degrade performance.

We also observe that training reward models solely on synthetic data generally underperforms compared to training on real data alone. Two factors may explain this. First, our evaluation is based on human-rated test datasets, and the real training data, which is also human-rated, is more likely to share a similar distribution with the test data. Second, the results presented in the main text were derived from reward models trained with an approximately equal mix of synthetic and real data. Although a pilot study using only synthetic data from the QA feedback dataset, amounting to roughly three times the volume of real data, showed some performance improvements, the process involved injecting significant human insights through ad hoc filtering rules. This introduces potential biases, so we opted not to include these results in our main experiments to ensure fairness.

H DATA GENERATION PROMPTS

Below we list the prompts used for all compared methods across three datasets. Specifically, for each dataset, we have one prompt used in RLAIF (for labeling two responses side-by-side), two prompts used in RLCD (one for generating the preferred response and the other for generating the less preferred response), and two prompts used in `RMBoost` (one for generating the first response and the other for preference-conditional generation of the second response). For `RMBoost`, we show the case where the first generated response is more preferred while the second one is less preferred. By switching the calling order of these two prompts (plus a few small word changes), we can obtain the case where the first response is less preferred than the second one.

H.1 QA FEEDBACK

H.1.1 RLAIF

Listing 1: Instruction used by RLAIF for QA Feedback Dataset

```
You work as a professor in a top-tier university. Your goal is to teach
students how to respond to a complex question given a set of Wikipedia
passages as context. In this task, you will be presented with a question,
 a set of Wikipedia passages, a reference response, and two candidate
responses that suppose to answer the given question. Your goal is to
compare these two candidate responses from a set of evaluation aspects
and decide which one is better for each evaluation aspect.

<task_description>
Below you will first see a guideline with detailed evaluation aspects of
the response. Then, you are presented with the question, the set of
Wikipedia passages, a reference response, and two candidate responses.
After that, for each aspect, please judge if one candidate response is
better than the other. Finally, you need to give an overall
recommendation on which candidate response is better. Think about your
answers first before making the judgement.
</task_description>

<guideline>
A good response to the question should provide both answer(s) that
directly responds to the qeustion and crucial auxiliary information for
better comprehension of the answer(s). We consider auxiliary information
as crucial if it is used in the reference response. Additionally, all
information in a good response should be factually consistent with (i.e.,
 grounded in) the passages. Note that the reference response is written
by a human with potentially different grounding passages, and thus, you
might find answers that can be found in the passages but are not included
 in the reference, which is considered as acceptable. On the other hand,
answers in the reference that cannot be found in or verifiable by the
passages are NOT expected to be in a good response. To conclude, all
answers are expected in a good response IF AND ONLY IF it can be found in
 the passages. Crucial auxiliary information is expected in a good
response IF AND ONLY IF it can be found in both the reference response
and the passages.

We will evaluate a response from the following aspects:
- (Relevance and Coherence): Whether the response contains irrelevant
information (e.g., neither an answer nor crucial auxiliary information)
and whether the response contains major grammar error (ignore minor typos
), is uninterpretable, contradicts to common sense, or is not coherent
with its context.
- (Factuality and Faithfulness): Whether the response is factually
consistent with the passages and contains information that is factually
verifiable. Common sense (e.g., "a bicycle has two wheels") doesn't need
to be verified. However, do not count knowledge only commonly known in a
specific region/community as common sense.
```

```
- (Completeness): Whether the response contains all needed information
including both the answer(s) that directly responds to the qeustion and
crucial auxiliary information mentioned in the reference response.
</guideline>

Below is the question.
<question>
[Question]
</question>

Below is the Wikipedia passages as the context for answering the above
question.
<passages>
[Passage]
</passages>

Below is the reference response.
<reference_response>
[Reference Response]
</reference_response>

Below is the first candidate response.
<first_response>
[First Response]
</first_response>

Below is the second candidate response.
<second_response>
[Second Response]
</second_response>

Read the question, passages, reference response, and two candidate
responses carefully. Review the above task description and guideline.
Please briefly recite your tasks back to me in your own words. For each
aspect, first decide if one candidate response is better than the other
and then write a one sentence explanation on why the selected response is
 better than the other one or why two responses are about the same. For
each aspect, put the decision in <ASPECT_NAME_answer></ASPECT_NAME_answer
> tags and the explanation in <ASPECT_NAME_explanation></
ASPECT_NAME_explanation> tags. Note here the ASPECT_NAME in the tags
should be replaced with the real aspect name (i.e. one of ["Relevance and
 Coherence", "Factuality and Faithfulness", "Completeness"]). Finally,
make an overall decision on which candidate response is better. Put the
overall decision in <final_answer></final_answer> tags and the
explanation in <final_explanation></final_explanation> tags. All
decisions should be in ["first_response", "second_response", "the_same"].
```

H.1.2 RLCD

Listing 2: Instruction used by RLCD for generating good QA Feedback response

```
You work as a professor in a top-tier university. You will be presented
with the question, a set of Wikipedia passages, a reference response, and
 aim to produce another good response to the question.

<guideline>
A good response to the question should provide both answer(s) that
directly responds to the qeustion and crucial auxiliary information for
better comprehension of the answer(s). We consider auxiliary information
as crucial if it is used in the reference response. Additionally, all
information in a good response should be factually consistent with (i.e.,
 grounded in) the passages. Note that the reference response is written
by a human with potentially different grounding passages, and thus, you
might find answers that can be found in the passages but are not included
```

```
 in the reference, which is considered as acceptable. On the other hand,
answers in the reference that cannot be found in or verifiable by the
passages are NOT expected to be in a good response. To conclude, all
answers are expected in a good response IF AND ONLY IF it can be found in
 the passages. Crucial auxiliary information is expected in a good
response IF AND ONLY IF it can be found in both the reference response
and the passages.

A good response should have the following qualities:
- (Relevance and Coherence): The response should not contain irrelevant
information (e.g., neither an answer nor crucial auxiliary information)
and the response should not contain major grammar error (ignore minor
typos). The response should be interpretable, follows common sense, and
coherent with its context.
- (Factuality and Faithfulness): The response should be consistent with
the passages and contains information that is factually verifiable.
Common sense (e.g., "a bicycle has two wheels") doesn't need to be
verified. However, the response should not count knowledge only commonly
known in a specific region/community as common sense.
- (Completeness): The response should contain all needed information
including both the answer(s) that directly responds to the qeustion and
crucial auxiliary information mentioned in the reference response.
</guideline>

Below is the question.
<question>
[Question]
</question>

Below is the Wikipedia passages as the context for answering the above
question.
<passages>
[Passage]
</passages>

Below is the reference response.
<reference_response>
[Reference Response]
</reference_response>

Read the question, passages, and reference response carefully. Review the
 above guideline. Think about how to accomplish the task step by step
before you reply. Put your generated good response in <response></
response> tags.
```

Listing 3: Instruction used by RLCD for generating bad QA Feedback response

```
You work as a professor in a top-tier university. You will be presented
with the question, a set of Wikipedia passages, a reference response, and
 aim to produce another bad response to the question.

<guideline>
A bad response to the question misses the answer(s) that directly
responds to the qeustion or crucial auxiliary information for better
comprehension of the answer(s). We consider auxiliary information as
crucial if it is used in the reference response. Additionally, some
information in the bad response may be factually inconsistent with (i.e.,
 not grounded in) the passages. Note that the reference response is
written by a human with potentially different grounding passages, and
thus, you might find answers that can be found in the passages but are
not included in the reference, which is considered as acceptable. On the
other hand, answers in the reference that cannot be found in or
verifiable by the passages are NOT expected to be in a good response.

A bad response will have at least one of the following qualities:
```

```
- (Bad Relevance and Coherence): The response contains irrelevant
information (e.g., neither an answer nor crucial auxiliary information)
or contains major grammar error (ignore minor typos). The response is
uninterpretable, contradicts to common sense, or is not coherent with its
 context.
- (Bad Factuality and Faithfulness): The response is not factually
consistent with the passages and contains information that is not
factually verifiable. Common sense (e.g., "a bicycle has two wheels")
doesn't need to be verified. However, knowledge only commonly known in a
specific region/community is not considered as common sense.
- (Bad Completeness): The response does not contain all needed
information including both the answer(s) that directly responds to the
qeustion and crucial auxiliary information mentioned in the reference
response.
</guideline>

Below is the question.
<question>
[Question]
</question>

Below is the Wikipedia passages as the context for answering the above
question.
<passages>
[Passage]
</passages>

Below is the reference response.
<reference_response>
[Reference Response]
</reference_response>

Read the question, passages, and reference response carefully. Review the
 above guideline. Think about how to accomplish the task step by step
before you reply. Put your generated bad response in <response></response
> tags.
```

H.1.3 RMBoost

Listing 4: Instruction used by RMBoost for generating the first QA Feedback response

```
You work as a professor in a top-tier university. Your goal is to teach
students how to respond to a complex question given a set of Wikipedia
passages as context. You will be presented with the question, a set of
Wikipedia passages, a reference response, and aim to produce another good
 response to the question.

<task_description>
Below you will first see a guideline with detailed evaluation aspects of
the response. Then, you are presented with the question, the set of
Wikipedia passages, and a reference response. After that, you need to
generate a good response to the question. This good response should NOT
directly copy the reference response but is roughly of the same length as
 the reference response.
</task_description>

<guideline>
A good response to the question should provide both answer(s) that
directly responds to the qeustion and crucial auxiliary information for
better comprehension of the answer(s). We consider auxiliary information
as crucial if it is used in the reference response. Additionally, all
information in a good response should be factually consistent with (i.e.,
 grounded in) the passages. Note that the reference response is written
by a human with potentially different grounding passages, and thus, you
```

```
might find answers that can be found in the passages but are not included
 in the reference, which is considered as acceptable. On the other hand,
answers in the reference that cannot be found in or verifiable by the
passages are NOT expected to be in a good response. To conclude, all
answers are expected in a good response IF AND ONLY IF it can be found in
 the passages. Crucial auxiliary information is expected in a good
response IF AND ONLY IF it can be found in both the reference response
and the passages.

We will evaluate a response from the following aspects:
- (Relevance and Coherence): Whether the response contains irrelevant
information (e.g., neither an answer nor crucial auxiliary information)
and whether the response contains major grammar error (ignore minor typos
), is uninterpretable, contradicts to common sense, or is not coherent
with its context.
- (Factuality and Faithfulness): Whether the response is factually
consistent with the passages and contains information that is factually
verifiable. Common sense (e.g., "a bicycle has two wheels") doesn't need
to be verified. However, do not count knowledge only commonly known in a
specific region/community as common sense.
- (Completeness): Whether the response contains all needed information
including both the answer(s) that directly responds to the qeustion and
crucial auxiliary information mentioned in the reference response.
</guideline>

Below is the question.
[Question]
{input_question}
</question>

Below is the Wikipedia passages as the context for answering the above
question.
<passages>
[Passage]
</passages>

Below is the reference response.
<reference_response>
[Reference Response]
</reference_response>

Read the question, passages, and reference response carefully. Review the
 above task description and guideline. Think about how to accomplish the
task step by step before you reply. Put your generated response in <
response></response> tags.
```

Listing 5: Instruction used by RMBoost for generating the second (less preferred) QA Feedback response

```
You work as a professor in a top-tier university. Your goal is to teach
students how to answer a complex question given a set of Wikipedia
passages. You will be presented with the question, a set of Wikipedia
passages, a reference response, a good response, and aim to produce
another response worse than the given good response for some aspects.

<task_description>
Below you will first see a guideline with detailed evaluation aspects of
the response. Then, you are presented with the question, the set of
Wikipedia passages, a reference response, and a good response. After that
, you need to accomplish the following steps:
Step 1: Select a few aspects in the guideline.
Step 2: Generate another response that is worse than the good response in
 terms of above selected aspects.
</task_description>
```

```
<guideline>
A good response to the question should provide both answer(s) that
directly responds to the qeustion and crucial auxiliary information for
better comprehension of the answer(s). We consider auxiliary information
as crucial if it is used in the reference response. Additionally, all
information in a good response should be factually consistent with (i.e.,
 grounded in) the passages. Note that the reference response is written
by a human with potentially different grounding passages, and thus, you
might find answers that can be found in the passages but are not included
 in the reference, which is considered as acceptable. On the other hand,
answers in the reference that cannot be found in or verifiable by the
passages are NOT expected to be in a good response. To conclude, all
answers are expected in a good response IF AND ONLY IF it can be found in
 the passages. Crucial auxiliary information is expected in a good
response IF AND ONLY IF it can be found in both the reference response
and the passages.

We will evaluate a response from the following aspects:
- (Relevance and Coherence): Whether the response contains irrelevant
information (e.g., neither an answer nor crucial auxiliary information)
and whether the response contains major grammar error (ignore minor typos
), is uninterpretable, contradicts to common sense, or is not coherent
with its context.
- (Factuality and Faithfulness): Whether the response is factually
consistent with the passages and contains information that is factually
verifiable. Common sense (e.g., "a bicycle has two wheels") doesn't need
to be verified. However, do not count knowledge only commonly known in a
specific region/community as common sense.
- (Completeness): Whether the response contains all needed information
including both the answer(s) that directly responds to the qeustion and
crucial auxiliary information mentioned in the reference response.
</guideline>

Below is the question.
<question>
[Question]
</question>

Below is the Wikipedia passages as the context for answering the above
question.
<passages>
[Passage]
</passages>

Below is the reference response.
<reference_response>
[Reference Response]
</reference_response>

Below is the good response.
<good_response>
[First Generated Good Response]
</good_response>

Read the question, passages, reference response, and the good response
carefully. Review the above task description and guideline. Think about
how to accomplish the task step by step before you reply. Put your
generated bad response in <response></response> tags.
```

H.2 Ultra Feedback

H.2.1 RLAIF

Listing 6: Instruction used by RLAIF for Ultra Feedback Dataset

```
You are a large language model researcher. Your goal is to train a
language model that follows the user input instruction with different
system prompts. In this task, you will be presented with a user input
instruction, a system prompt, and two candidate responses that suppose to
 follow the user input instruction. Your goal is to compare these two
candidate responses from a set of evaluation aspects and decide which one
 is better for each evaluation aspect.

<task_description>
Below you will first see a guideline with detailed evaluation aspects of
the response. Then, you are presented with the instruction, the system
prompt, and two candidate responses. After that, for each aspect, please
judge if one candidate response is better than the other. Finally, you
need to give an overall recommendation on which candidate response is
better. Think about your answers first before making the judgement.
</task_description>

<guideline>
We will evaluate a response from the following aspects:
- (Honesty): The assistant should be honest about whether it knows the
answer and express its uncertainty explicitly. Be confident on questions
it knows well and be modest on those it is unfamiliar with. Use weakeners
 such as 'I guess', 'I suppose', 'probably', and 'perhaps' to express
uncertainty, and feel free to answer 'I donâĂŹt know' if necessary.
- (Truthfulness): The assistant should answer truthfully and be faithful
to factual knowledge as well as given contexts, never making up any new
facts that arenâĂŹt true or cannot be grounded in the instruction.
- (Faithful to input): The article should be faithful to the original
press release without adding unsupported information or inaccurate
statements.
- (Helpfulness): The assistant should provide users with accurate,
relevant, and up-to-date information, ensuring that the content is
positive, interesting, engaging, educational, and helpful.
- (Verbalized Calibration): The assistant should express its confidence
as a scalar at the end of the response. The confidence level indicates
the degree of certainty it has about its answer and is represented as a
percentage.
</guideline>

Below is the system prompt.
<system_prompt>
[System Prompt]
</system_prompt>

Below is the user input instruction.
<instruction>
[Instruction]
</instruction>

Below is the first candidate response.
<first_response>
[First Response]
</first_response>

Below is the second candidate response.
<second_response>
[Second Response]
</second_response>

Read the user input instruction, system prompt, and two candidate
responses carefully. Review the above task description and guideline.
Please briefly recite your tasks back to me in your own words. For each
aspect, first decide if one candidate response is better than the other
```

```
and then write a one sentence explanation on why the selected response is
 better than the other one or why two responses are about the same. For
each aspect, put the decision in <ASPECT_NAME_answer></ASPECT_NAME_answer
> tags and the explanation in <ASPECT_NAME_explanation></
ASPECT_NAME_explanation> tags. Note here the ASPECT_NAME in the tags
should be replaced with the real aspect name (i.e. one of ["Honesty", "
Truthfulness", "Faithful to input", "Helpfulness", "Verbalized
Calibration"]). Finally, make an overall decision on which candidate
response is better. Put the overall decision in <final_answer></
final_answer> tags and the explanation in <final_explanation></
final_explanation> tags. All decisions should be in ["first_response", "
second_response", "the_same"].
```

H.2.2 RLCD

Listing 7: Instruction used by RLCD for generating good Ultra Feedback response

```
You are a large language model researcher. Your goal is to train a good
model that follows the user input instruction with different system
prompts. You will be presented with the instruction, the system prompt,
and aim to produce one good response.

<guideline>
A good response should come from an assistant that has the following
qualities:
- (Honesty): The assistant should be honest about whether it knows the
answer and express its uncertainty explicitly. Be confident on questions
 it knows well and be modest on those it is unfamiliar with. Use weakeners
 such as 'I guess', 'I suppose', 'probably', and 'perhaps' to express
uncertainty, and feel free to answer 'I donâĂŹt know' if necessary.
- (Truthfulness): The assistant should answer truthfully and be faithful
to factual knowledge as well as given contexts, never making up any new
facts that arenâĂŹt true or cannot be grounded in the instruction.
- (Faithful to input): The article should be faithful to the original
press release without adding unsupported information or inaccurate
statements.
- (Helpfulness): The assistant should provide users with accurate,
relevant, and up-to-date information, ensuring that the content is
positive, interesting, engaging, educational, and helpful.
- (Verbalized Calibration): The assistant should express its confidence
as a scalar at the end of the response. The confidence level indicates
the degree of certainty it has about its answer and is represented as a
percentage.
</guideline>

Below is the system prompt.
<system_prompt>
[System Prompt]
</system_prompt>

Below is the user input instruction.
<instruction>
[Instruction]
</instruction>

Read the system prompt and instruction carefully. Review the above
guideline. Think about your output response first before you respond. Put
 your generated good response in <response></response> tags.
```

Listing 8: Instruction used by RLCD for generating bad Ultra Feedback response

```
You are a large language model researcher. Your goal is to train a good
model that follows the user input instruction with different system
```

```
prompts. You will be presented with the instruction, the system prompt,
and aim to produce one bad response that a good LLM will not generate.

<guideline>
A bad response should come from an assistant that has at least one of the
 following qualities:
- (Bad Honesty): The assistant is not honest about whether it knows the
answer and fails to express its uncertainty explicitly. The assistant is
over-confident on questions it doesn't know well and is not modest on
those it is unfamiliar with.
- (Bad Truthfulness): The assistant doesn't answer truthfully and is not
faithful to factual knowledge as well as given contexts. The assistant
makes up some new facts that arenâĂŹt true or cannot be grounded in the
instruction.
- (Not Faithful to input): The assistant produces response that is not
faithful to the original press release or adds unsupported information
and inaccurate statements.
- (Not Helpful): The assistant doesn't provide users with accurate,
relevant, and up-to-date information. The assistant fails to output
content that is positive, interesting, engaging, educational, and helpful
.
- (Not Verbalized Calibration): The assistant fails to express its
confidence as a scalar at the end of the response. The confidence level
indicates the degree of certainty it has about its answer and is
represented as a percentage.
</guideline>

Below is the system prompt.
<system_prompt>
[System Prompt]
</system_prompt>

Below is the user input instruction.
<instruction>
[Instruction]
</instruction>

Read the system prompt and instruction carefully. Review the above
guideline. Think about your output response first before you respond. Put
 your generated bad response in <response></response> tags.
```

H.2.3 RMBoost

Listing 9: Instruction used by RMBoost for generating the first Ultra Feedback response

```
You are a large language model researcher. Your goal is to train a good
model that follows the user input instruction with different system
prompts. You will be presented with the instruction, the system prompt,
and aim to produce one good response.

<task_description>
Below you will first see a guideline with detailed evaluation aspects of
the response. Then, you are presented with the system prompt, the
instruction. After that, you need to generate a good response to the
question.
</task_description>

<guideline>
A response is evaluated from the following aspects:
- (Honesty): The assistant should be honest about whether it knows the
answer and express its uncertainty explicitly. Be confident on questions
it knows well and be modest on those it is unfamiliar with. Use weakeners
 such as 'I guess', 'I suppose', 'probably', and 'perhaps' to express
uncertainty, and feel free to answer 'I donâĂŹt know' if necessary.
```

```
- (Truthfulness): The assistant should answer truthfully and be faithful
to factual knowledge as well as given contexts, never making up any new
facts that arenâĂŹt true or cannot be grounded in the instruction.
- (Faithful to input): The article should be faithful to the original
press release without adding unsupported information or inaccurate
statements.
- (Helpfulness): The assistant should provide users with accurate,
relevant, and up-to-date information, ensuring that the content is
positive, interesting, engaging, educational, and helpful.
- (Verbalized Calibration): The assistant should express its confidence
as a scalar at the end of the response. The confidence level indicates
the degree of certainty it has about its answer and is represented as a
percentage.
</guideline>

Below is the system prompt.
<system_prompt>
[System Prompt]
</system_prompt>

Below is the user input instruction.
<instruction>
[Instruction]
</instruction>

Read the system prompt and instruction carefully. Review the above task
description and guideline. Think about your output response first before
you respond. Put your generated response in <response></response> tags.
```

Listing 10: Instruction used by RMBoost for generating the second (less preferred) Ultra Feedback response

```
You are a large language model researcher. Your goal is to train a good
model that follows the user input instruction with different system
prompts. You will be presented with the instruction, the system prompt, a
 good response and aim to produce another response worse than the given
good response for some aspects.

<task_description>
Below you will first see a guideline with detailed evaluation aspects of
the response. Then, you are presented with the system prompt, the
instruction, and a good response. After that, you need to accomplish the
following steps:
Step 1: Select a few aspects in the guideline.
Step 2: Generate another response that is worse than the good response in
 terms of above selected aspects.
</task_description>

<guideline>
A response is evaluated from the following aspects:
- (Honesty): The assistant should be honest about whether it knows the
answer and express its uncertainty explicitly. Be confident on questions
it knows well and be modest on those it is unfamiliar with. Use weakeners
 such as 'I guess', 'I suppose', 'probably', and 'perhaps' to express
uncertainty, and feel free to answer 'I donâĂŹt know' if necessary.
- (Truthfulness): The assistant should answer truthfully and be faithful
to factual knowledge as well as given contexts, never making up any new
facts that arenâĂŹt true or cannot be grounded in the instruction.
- (Faithful to input): The article should be faithful to the original
press release without adding unsupported information or inaccurate
statements.
- (Helpfulness): The assistant should provide users with accurate,
relevant, and up-to-date information, ensuring that the content is
positive, interesting, engaging, educational, and helpful.
```

```
- (Verbalized Calibration): The assistant should express its confidence
as a scalar at the end of the response. The confidence level indicates
the degree of certainty it has about its answer and is represented as a
percentage.
</guideline>

Below is the system prompt.
<system_prompt>
[System Prompt]
</system_prompt>

Below is the user input instruction.
<instruction>
[Instruction]
</instruction>

Below is the good response.
<good_response>
[First Generated Good Response]
</good_response>

Read the system prompt and instruction carefully. Review the above task
description and guideline. Think about how to accomplish the task step by
 step before you respond. Put your generated bad response in <response></
response> tags.
```

H.3 TLDR SUMMARIZATION

H.3.1 RLAIF

Listing 11: Instruction used by RLAIF for TLDR Summarization Dataset

```
You work as a professor in a top-tier university. You aim to teach
students how to summarize a reddit post. In this task, you will be
presented with a post and two candidate summaries that suppose to
summarize the given post. Your goal is to compare these two candidate
summaries from a set of evaluation aspects and decide which one is better
 for each evaluation aspect.

<task_description>
Below you will first see a guideline with detailed evaluation aspects of
the summary. Then, you are presented with the post and two candidate
summaries. After that, for each aspect, please judge if one candidate
summary is better than the other. Finally, you need to give an overall
recommendation on which candidate summary is better. Think about your
answers first before making the judgement.
</task_description>

<guideline>
A good summary is a shorter piece of text that accomplishes the same
purpose and conveys the same information as the original post.

We will evaluate a summary from the following aspects:
- (Coherence): A summary is coherent if, when read by itself it's easy to
 understand and free of English errors. A summary is not coherent if it
is difficult to understand what the summary is trying to say. It's more
important that the summary is understandable than it being free of
grammar errors.
- (Accuracy): A summary is accurate if it doesn't say things that aren't
in the post, it doesn't mix up people, and generally is not misleading.
If the summary says anything that is not mentioned in the post or
contradicts something in the post, this summary is not accurate enough.
- (Coverage): A summary has good coverage if it mentions the main
information from the post that's important to understand the situation
```

```
described in the post. A summary has poor coverage if someone reading
only the summary would be missing several important pieces of information
 about the situation in the post.
</guideline>

Below is the post.
<post>
[Post]
</post>

Below is the first candidate summary.
<first_summary>
[First Summary]
</first_summary>

Below is the second candidate summary.
<second_summary>
[Second Summary]
</second_summary>

Read the post and two candidate summaries carefully. Review the above
task description and guideline. Please briefly recite your tasks back to
me in your own words. For each aspect, first decide if one candidate
summary is better than the other and then write a one sentence
explanation on why the selected summary is better than the other one or
why two summaries are about the same. For each aspect, put the decision
in <ASPECT_NAME_answer></ASPECT_NAME_answer> tags and the explanation in
<ASPECT_NAME_explanation></ASPECT_NAME_explanation> tags. Note here the
ASPECT_NAME in the tags should be replaced with the real aspect name (i.e
. one of ["Coherence", "Accuracy", "Coverage"]). Finally, make an overall
 decision on which candidate summary is better. Put the overall decision
in <final_answer></final_answer> tags and the explanation in <
final_explanation></final_explanation> tags. All decisions should be in
["first_summary", "second_summary", "the_same"].
```

H.3.2 RLCD

Listing 12: Instruction used by RLCD for generating good TLDR Summarization response

```
You work as a professor in a top-tier university. Your goal is to teach
students how to summarize a reddit post. A good summary is a shorter
piece of text that accomplishes the same purpose and conveys the same
information as the original post. You will be presented with the post and
 aim to produce a good response as the summary.

<guideline>
A good summary should have the following qualities:
- (Coherence): A good summary is coherent if, when read by itself it's
easy to understand and free of English errors. It's more important that
the good summary is understandable than it being free of grammar errors.
- (Accuracy): A good summary is accurate if it doesn't say things that
aren't in the post, it doesn't mix up people, and generally is not
misleading. If the summary says anything that is not mentioned in the
post or contradicts something in the post, this summary is not accurate
enough.
- (Coverage): A good summary has good coverage if it mentions the main
information from the post that's important to understand the situation
described in the post. A summary has poor coverage if someone reading
only the summary would be missing several important pieces of information
 about the situation in the post.
</guideline>

Below is the post.
<post>
```

```
[Post]
<post>

Read the post and guideline carefully. Think about your output summary
first before you respond. Put your generated good response in <response
></response> tags.
```

Listing 13: Instruction used by RLCD for generating bad TLDR Summarization response

```
You work as a professor in a top-tier university. Your goal is to teach
students how to summarize a reddit post. A good summary is a shorter
piece of text that accomplishes the same purpose and conveys the same
information as the original post. The bad summary fails to accomplish
this goal and thus you will teach student not to produce it. Below, you
will be presented with the post and aim to produce a bad response as the
summary.

<guideline>
A bad summary will have at least one of the following qualities:
- (Bad Coherence): A bad summary is not coherent when read by itself it's
 hard to understand and has some English errors. A bad summary is not
coherent as it is difficult to understand what the summary is trying to
say.
- (Bad Accuracy): A bad summary is not accurate as it says something that
 is not mentioned in the post or contradicts something in the post.
- (Bda Coverage): A bad summary has poor coverage when someone reading
only the summary would be missing several important pieces of information
 about the situation in the post.
</guideline>

Below is the post.
<post>
[Post]
<post>

Read the post and guideline carefully. Think about your output summary
first before you respond. Put your generated bad response in <response></
response> tags.
```

H.3.3 RMBoost

Listing 14: Instruction used by RMBoost for generating the first TLDR Summarization response

```
You work as a professor in a top-tier university. Your goal is to teach
students how to summarize a reddit post. A good summary is a shorter
piece of text that accomplishes the same purpose and conveys the same
information as the original post. You will be presented with the post and
 aim to produce a good response as the summary.

<task_description>
Below you will first see a guideline with detailed evaluation aspects of
the summary. Then, you are presented with the post. After that, you need
to generate a good summary of the post as the response.
</task_description>

<guideline>
A summary is evaluated from the following aspects:
- (Coherence): A summary is coherent if, when read by itself it's easy to
 understand and free of English errors. A summary is not coherent if it
is difficult to understand what the summary is trying to say. It's more
important that the summary is understandable than it being free of
grammar errors.
- (Accuracy): A summary is accurate if it doesn't say things that aren't
in the post, it doesn't mix up people, and generally is not misleading.
```

```
If the summary says anything that is not mentioned in the post or
contradicts something in the post, this summary is not accurate enough.
- (Coverage): A summary has good coverage if it mentions the main
information from the post that's important to understand the situation
described in the post. A summary has poor coverage if someone reading
only the summary would be missing several important pieces of information
 about the situation in the post.
</guideline>

Below is the post.
<post>
[Post]
<post>

Read the post carefully. Review the above task description and guideline.
 Think about your output summary first before you respond. Put your
generated response in <response></response> tags.
```

Listing 15: Instruction used by RMBoost for generating the second (less preferred) TLDR Summarization response

```
You work as a professor in a top-tier university. Your goal is to teach
students how to summarize a reddit post. A good summary is a shorter
piece of text that accomplishes the same purpose and conveys the same
information as the original post. You will be presented with the post, a
good response, and aim to produce another response worse than the given
good response for some aspects.

<task_description>
Below you will first see a guideline with detailed evaluation aspects of
the summary. Then, you are presented with the post and a good response.
After that, you need to accomplish the following steps:
Step 1: Select a few aspects in the guideline.
Step 2: Generate another response that is worse than the first response
in terms of above selected aspects.
</task_description>

<guideline>
A summary is evaluated from the following aspects:
- (Coherence): A summary is coherent if, when read by itself it's easy to
 understand and free of English errors. A summary is not coherent if it
is difficult to understand what the summary is trying to say. It's more
important that the summary is understandable than it being free of
grammar errors.
- (Accuracy): A summary is accurate if it doesn't say things that aren't
in the post, it doesn't mix up people, and generally is not misleading.
If the summary says anything that is not mentioned in the post or
contradicts something in the post, this summary is not accurate enough.
- (Coverage): A summary has good coverage if it mentions the main
information from the post that's important to understand the situation
described in the post. A summary has poor coverage if someone reading
only the summary would be missing several important pieces of information
 about the situation in the post.
</guideline>

Below is the post.
<post>
[Post]
<post>

Below is the good response.
<good_response>
[Generated First Good Response]
<good_response>
```

```
Read the post carefully. Review the above task description and guideline.
 Think about how to accomplish the task step by step before you reply.
Put your generated bad summary in <response></response> tags.
```