# RS-DPO: A Hybrid Rejection Sampling and Direct Preference Optimization Method for Alignment of Large Language Models

**Saeed Khaki**
Amazon
sakhaki@amazon.com

**JinJin Li**
Amazon
jinjinli@amazon.com

**Lan Ma**
Amazon
mamlm@amazon.com

**Liu Yang**
Amazon
liuyanga@amazon.com

**Prathap Ramachandra**
Amazon
prathara@amazon.com

## Abstract

Reinforcement learning from human feedback (RLHF) has been extensively employed to align large language models with user intent. However, proximal policy optimization (PPO) based RLHF is occasionally unstable requiring significant hyperparameter finetuning, and computationally expensive to maximize the estimated reward during alignment. Recently, direct preference optimization (DPO) is proposed to address those challenges. However, DPO often relies on contrastive responses generated from human annotator and alternative LLM, instead of the policy model, limiting the effectiveness of the RLHF. In this paper, we addresses both challenges by systematically combining rejection sampling (RS) and DPO. Our proposed method, RS-DPO, initiates with the development of a supervised fine-tuned policy model (SFT). A varied set of k responses per prompt are sampled directly from the SFT model. RS-DPO identifies pairs of contrastive samples based on their reward distribution. Finally, we apply DPO with the contrastive samples to align the model to human preference. Our experiments indicate that our proposed method effectively fine-tunes LLMs with limited resource environments, leading to improved alignment with user intent. Furthermore, it outperforms existing methods, including RS, PPO, and DPO.

## 1 Introduction

State-of-the-art (SOTA) LLMs such as GPT-4 (OpenAI, 2023), LLaMa (Touvron et al., 2023) etc., are trained with several stages. With pre-training and supervised instruction tuning, LLMs learns to follow specific instructions to complete various tasks with zero-shot or few-shot prompts (Chowdhery et al., 2022). To further improve the LLMs' intelligence as close as to human and ensure a more helpful and harmless model, alignment is important as the last-mile LLM training procedure (Ziegler et al., 2019; Stiennon et al., 2020b; Bai et al., 2022; Ouyang et al., 2022). Reinforcement learning with human feedback (RLHF) (Christiano et al., 2017) is the most adopted approach for alignment training and it usually involves training a reward model with human preference datasets which optimizes a reward function based on the human-annotated preference. Then LLMs are fine-tuned to learn to maximize the reward of their responses using reinforcement learning algorithms, including proximal policy optimization (PPO) (Schulman et al., 2017), REINFORCE (Williams, 2004), and similar variants. While PPO is used by SOTA LLMs due to its ease of use and good performance, training with PPO has few limitations, including complexity of training multiple LLMs, and sampling from policy model in training loop, high GPU memory requirement with hosting multiple LLMs during training, and sensitivity to training data and reward models.

To make RLHF training more efficient, there are methods proposed from different perspective. In order to reduce the preference data effort by human annotation, (Lee et al., 2023) and (Tunstall et al., 2023) proposed to train the LLM to align to the LLM's preference rating in order to save human effort. (Santacroce et al., 2023) proposed a combined strategy to merge SFT and reward models as well as in PPO with LoRA selection in order to reduce latency and memory footprint. (Dong et al., 2023; Gulcehre et al., 2023) used reward model to select ranked high-reward good samples to supervise fine-tune the models and iteratively repeating this process yield good results. To reduce the memory and save training resources, (Rafailov et al., 2023) proposed the direct preference optimization (DPO) to remove the need of training reward model, and directly optimize the policy model using a simple classification to maximize the difference between likelihood of human preference pairs. This method proves equivalent performance by implicitly maximize the reward. However, it is mainly trained on human preference data to learn the alignment,



*Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1665–1680

instead of sampling the policy model's response for optimization. LLaMa2 (Touvron et al., 2023) adopts several rounds of rejection sampling to select the best samples from $k$ model-generated samples for fine-tuning before PPO in order to boost the model performance. But rejection sampling only selects the best samples instead of preference pairs, with low data usage efficiency. RSO (Liu et al., 2023) proposes to generate preference data from the target optimal policy using rejection sampling, enabling a more accurate estimation of the optimal policy. Compared to RSO, our proposed method (RS-DPO) directly employs a point-wise reward model for response ranking and optimization, utilizing logistic loss exclusively during policy optimization. Unlike RSO's approach of statistical rejection sampling and tournament ranking for response generation and selection, RS-DPO generates a fixed number of responses per prompt and relies on computing reward gaps between responses for preference data generation, resulting in reduced computational expense. Additionally, while RSO lacks evaluation on standard alignment benchmarks and comparison against PPO, RS-DPO demonstrates its effectiveness against other RLHF methods on such benchmarks.

In this work, we propose RS-DPO method for RLHF training that combines the advantages of existing efficient methods, including offline preference data generation using rejection sampling, and using DPO in order to reduce the training GPU memory consumption. Specifically, RS-DPO generates responses from the large language model directly, and leverages rejection sampling (RS) to sample synthetic preference pairs based on the reward distribution of LLMs responses. Then, it uses the generated preference pairs for alignment with DPO. The main contributions of our proposed RLHF training method can be summarized as follows: (1) RS-DPO demonstrates stability and robustness against variations in the reward model quality, consistently outperforming existing methods like DPO, PPO and RS. (2) In contrast to the rejection sampling approach that focuses solely on the best response among $k$ generated responses for alignment, RS-DPO selects pairs of contrastive samples based the reward distribution, thereby enhancing overall performance. (3) RS-DPO samples contrastive data directly from the SFT model, distinguishing itself from DPO which often relies on responses from alternative language models or human annotations. This approach contributes to the superior performance of RS-DPO. (4) Our proposed method is efficient, being less resource-intensive compared to PPO, making it practical for applications in limited resource environments.

## 2 Method

The aim of this study is to utilize reinforcement learning from human feedback (RLHF) to train a policy model with the purpose of aligning a large language model to user intent. As the pipeline shown in 1, our proposed method, RS-DPO, systematically combines RS and DPO. It starts by generating a diverse set of $k$ distinct responses for each prompt, selecting a pair of contrasting samples based on their reward distribution. Subsequently, the method employs DPO to enhance the performance of the language model (LLM), thereby achieving improved alignment. Our proposed method consists the following steps:

### 2.1 Supervised Fine-Tuning (SFT)

As a prerequisite to RLHF, this step involves fine-tuning a pre-trained LLM, $\pi$, using a dataset consisting of high-quality instruction and response pairs or chat data, denoted as $\mathcal{D}_{\text{sft}} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ (Ouyang et al., 2022; Wang et al., 2023a; Chung et al., 2022; Wang et al., 2022). Starting from a base LLM $\pi$, SFT maximizes the likelihood of response $y$ given prompt $x$ as defined in the Equation 1.

$$\mathcal{L}^{\text{SFT}} = \text{argmax} \sum_{(x,y)\in\mathcal{D}_{\text{sft}}} \log \pi(y|x) \qquad (1)$$

### 2.2 Reward Model Training (RM)

This step involves training a reward model to assess the quality of a response in accordance with human preferences, with a focus on desired downstream attributes like helpfulness and harmlessness (Wang et al., 2023a; Ouyang et al., 2022). The reward model, denoted as $R(x, y)$, takes a prompt $x$ and a response $y$, and maps them to a scalar value $r$. Let's assume that we have a preference dataset, denoted as $\mathcal{D}_{\text{RM}} = \{(x_1, y_{1l}, y_{1w}), \ldots, (x_n, y_{nl}, y_{nw})\}$, where $x$ represents the input prompt, and $y_l$ and $y_w$ are considered the worse and the better responses, respectively, as determined by human assessment. Reward model training uses ranked answers from $\mathcal{D}_{\text{RM}}$ to estimate the preference distribution $p$ as written in Equation 2 (Bradley and Terry, 1952).

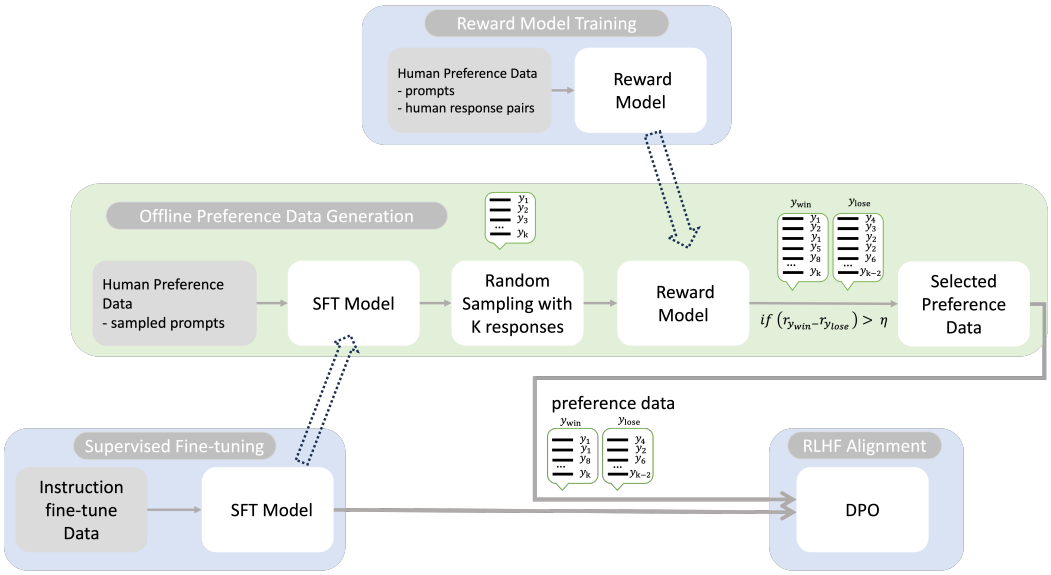


Figure 1: The pipeline of RS-DPO. Our proposed method systematically combines rejection sampling (RS) and direct preference optimization (DPO). We start by creating a SFT model and use it to generate a diverse set of $k$ distinct responses for each prompt. Then, it selects a pair of contrastive samples based on their reward distribution. Subsequently, the method employs DPO to enhance the performance of the language model (LLM), thereby achieving improved alignment.

$$p(y_w \succ y_l|x) = \frac{\exp\left(r(x, y_w)\right)}{\exp\left(r(x, y_w)\right) + \exp\left(r(x, y_l)\right)} \tag{2}$$

Subsequently, we can estimate the parameters of the reward model through maximum likelihood by maximizing the reward gap between $y_w$ and $y_l$, as illustrated in Equation 3 (Wang et al., 2023b; Rafailov et al., 2023).

$$R(x, y) = \text{argmin} \sum_{(x, y_l, y_w) \in \mathcal{D}_{\text{RM}}} - \log \sigma(r(x, y_w)) - (r(x, y_l)) \tag{3}$$

### 2.3 Preference Data Generation via Rejection Sampling (PDGRS)

The goal of this step is to create a synthetic preference pair dataset for our alignment task using the trained SFT and RM. Let's denote $\{x_1, \ldots, x_n\}$ as a set of sampled prompts from $\mathcal{D}_{\text{RM}}$ dataset. While it is possible to sample from other prompt datasets, it is crucial to ensure that our prompt sampling remains within the reward model's prompt distribution for optimal performance. We first generate $k$ distinct responses from $\mathcal{L}^{\text{SFT}}$ model for each prompt $x$. Then, we evaluate the quality of each response using our trained reward model $R(x, y)$. Finally, we compute the reward gap for all possible pairwise combinations of responses per prompt, $\binom{k}{2}$. If the reward gap surpasses a predefined threshold, we include the pair of responses in our synthetic preference dataset. The process of preference data generation is illustrated in Algorithm 1.

Since the preference data generation process generates responses from $\mathcal{L}_{\text{SFT}}$ model, it ensures that our RLHF is focused on aligning the $\mathcal{L}_{\text{SFT}}$ behaviour to the human preference rather than distilling knowledge from a larger model or human annotations. We term this process preference data generation via rejection sampling (PDGRS), as it involves evaluating each possible preference data triplet combination (superior and inferior responses), and discarding those with reward gaps below predefined threshold. In addition, our proposed preference data generation process bootstraps and substantially augments the quantity of preference data, compared to the initial static preference dataset $\mathcal{D}_{\text{RM}}$ used in the reward model training.

### 2.4 Direct Preference Optimization (DPO)

DPO fine-tunes $\mathcal{L}^{\text{SFT}}$ by directly optimizing the policy model on static preference data $(x, y_l, y_w)$, maximizing the likelihood of the preferred $y_w$ over $y_l$. This approach eliminates the necessity of fitting an explicit reward model by using the ratio of likelihood between the policy $\mathcal{L}^{\text{RL}}$ model and the original $\mathcal{L}^{\text{SFT}}$ model as an implicit reward signal

**Algorithm 1** Preference Data Generation via Rejection Sampling

**Result**:
$\mathcal{D}_{\text{P}} = \{(x, y_l, y_w)\}_{3m}$ : Preference dataset
**Input**:
$\{x_1, \ldots, x_n\}$ : Sample prompts from $\mathcal{D}_{\text{RM}}$
$\mathcal{L}^{\text{SFT}}$: SFT model
$R(x, y)$: Reward model
$\tau$: Temperature
$\eta$: Threshold for preference data selection
**for** $i = 1 : n$ **do**
  $(y_{i1}, \ldots, y_{ik}) \,|\, y_{ik} \sim \mathcal{L}^{\text{SFT}}(\cdot|x_i)$ ▷ generate $k$ responses from $\mathcal{L}^{\text{SFT}}$ model for prompt $x_i$
  $(r_{i1}, \ldots, r_{ik}) \,|\, r_{ij} = R(x_i, y_{ij})$ ▷ compute the reward for each of generated responses
  **for** $j = 1 : k$ **do**
    **for** $l = 1 : k$ **do**
      **if** $j == l$ **then**
        continue
      **end if**
      $r_{\text{gap}} = \sigma(\frac{r_{ij} - r_{il}}{\tau})$ ▷ compute the reward gap between the pair of responses $y_{il}$ and $y_{ij}$
      **if** $r_{\text{gap}} > \eta$ **then**
        $\mathcal{D}_{\text{P}} = \{\mathcal{D}_{\text{P}}; (x_i, y_{il}, y_{ij})\}$ ▷ append the accepted sample
      **end if**
    **end for**
  **end for**
**end for**

(Rafailov et al., 2023). During training process, DPO optimizes the objective function as written in Equation 4.

$$\mathcal{L}^{\text{RL}} = \text{argmax} \sum_{(x, y_l, y_w) \in \mathcal{D}_{\text{P}}} \log \sigma(\beta \log \frac{\mathcal{L}^{\text{RL}}(y_w|x)}{\mathcal{L}^{\text{SFT}}(y_w|x)} - \beta \log \frac{\mathcal{L}^{\text{RL}}(y_l|x)}{\mathcal{L}^{\text{SFT}}(y_l|x)}) \quad (4)$$

Our proposed method employs DPO on the synthetic preference dataset $\mathcal{D}_{\text{P}}$ generated in step 2.3 using PDGRS, in order to align the policy model with human preferences.

## 3 Experiments Details

This section outlines our experiments to demonstrate the effectiveness of our proposed RS-DPO method for the alignment task. We conduct all of our experiments on the Llama-2-7B LLM (Touvron et al., 2023) which is one of the state-of-the-art LLMs at 7B parameter scale. We perform supervised fine-tuning (SFT), reward modeling (RM), DPO training, and PPO training based on Huggingface Transformer Reinforcement Learning (TRL) library (von Werra et al., 2020). We utilize DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) for optimizing GPU memory and training speed. All experiments are conducted on 8 A100s GPUs with 40G memory per GPU.

### 3.1 Datasets

We use the following datasets in our experiments:

**Open Assistant:** Open Assistant (OASST1) (Köpf et al., 2023) is a multilingual human-generated conversation dataset ranked for quality. In our experiment, we utilize the highest quality partition based on quality ranking, comprising of 9k samples.

**Anthropic/HH-RLHF:** Anthropic released this dataset that includes 169.55k conversation pairs between humans and an AI assistant to train a helpful and safe AI assistant. This preference dataset has two subsets namely helpfulness and harmlessness (Bai et al., 2022; Ganguli et al., 2022). In our experiments, we only use a random sample of the helpfulness subset of the data with the size of roughly 10,300 samples.

**WebGPT:** WebGPT (Nakano et al., 2021) dataset includes long-form question answering preference dataset annotated by humans for reward modeling. After cleaning this dataset, we get 17,814 samples from this dataset.

### 3.2 Experimental Setup

We start our experiments by training a Llama-2-7B SFT model using the Open Assistant conversation dataset. We specifically choose this SFT dataset for two primary reasons: (1) the same SFT model is used across different preference datasets in RLHF. This helps to examine the influence of preference dataset on our proposed method, and (2) the utilization of high-quality chat data leads to the improved performance of SFT models (Dettmers et al., 2023). For SFT step, we employ linear learning rate schedule with starting learning rate of $2 \times 10^{-5}$, effective batch size of 64, number of epochs of 2, weight decay of 0.1, and a sequence length of 4096 tokens. We do not use LoRA (Hu et al., 2021) finetuning in the SFT step.

In the response generation step during the PDGRS (1) of our proposed method, we generate a total of $k = 16$ responses for each prompt, with the following decoding parameters: a maximum of 512 new tokens, a top-k value of 50, a top-p value of 0.98, and a sampling temperature of 1. We applied PDGRS on 10,300 samples from Anthropic/HH-RLHF, and 12,193 samples from WebGPT.

To assess the quality of generated responses, we employ the pythia-6.9B reward model developed by Open Assistant, denoted as pythia-6.9B-RM-OA in our experiments (OpenAssistant, 2023). This reward model is trained on a diverse set of datasets, including Open Assistant preference (Köpf et al., 2023), Anthropic (Bai et al., 2022; Ganguli et al., 2022), SHP (Ethayarajh et al., 2022), hellaswag (Zellers et al., 2019), WebGPT (Nakano et al., 2021), and summary pairs (Stiennon et al., 2020a). To control the impact of reward model preference data with our proposed method, we also trained a pythia-6.9B reward model using only WebGPT preference dataset, denoted as pythia-6.9B-RM-WG in our experiments. We use pythia-6.9B (Biderman et al., 2023) as a base model and train it for 1 epoch with learning rate of $1 \times 10^{-5}$ with linear learning rate schedule.

For DPO training in our experiments, we use cosine learning rate schedule with an initial learning rate of $1 \times 10^{-6}$, effective batch size of 64, number of epochs of 4, $\beta = 0.1$, and a sequence length of 4096 tokens. We use LoRA with rank $= 8$ to enable training Llama-2-7B models with limited GPU resources.

For PPO (Schulman et al., 2017) training in our experiments, we use LoRA with rank $= 8$ and 8-bit quantization for both policy and reward models. We adopt effective batch size of 64, learning rate of $2 \times 10^{-5}$, and Kullback-Leibler (KL) coefficient of 0.2. We train the policy model between 150-200 steps to converge.

### 3.3 Evaluation

Assessing alignment to human preference is challenging, but recent developments have introduced specialized benchmarks like MT-Bench (Zheng et al., 2023) and AlpacaEval (Li et al., 2023) to address this issue. These benchmarks leverage strong LLM judges like GPT-4, providing a score that strongly correlates with human preference ratings. We use the following benchmarks to evaluate model's performance on instruction following and alignment to user intent:

**MT-Bench:** MT-Bench evaluation is based on GPT-4 judgement and achieves over 80% agreement with human preference. MT-bench is designed to test multi-turn conversation and instruction-following ability of LLMs, covering 8 common categories including writing, roleplay, extraction, reasoning, math, coding, knowledge I (STEM), and knowledge II (humanities/social science). MT-Bench has 10 multi-turn questions for each category, and GPT-4 rates each turn's response on a scale of 1-10, with the final score being the mean over two turns (Zheng et al., 2023).

**AlpacaEval:** It is an LLM-based automatic evaluation judged by GPT-4, where it measures the pairwise win-rate against a baseline model (text-davinci-003). We use 300 questions mostly focused on helpfulness from this benchmark in our evaluations (Li et al., 2023).

## 4 Results and Ablations

This section presents our main results. We show sample model completions in appendix D. To comprehensively assess the effectiveness of our proposed method, we employ a comparative analysis of various preference data generation policies. These policies guide the selection of the superior model response, denoted as $y_w$, and the inferior model response, denoated as $y_l$, from a set of $k$ generated answers. The following preference data generation policies are considered:

**Best-vs-worst:** This policy ranks the $k$ responses according to their respective rewards and selects the response with the highest reward as $y_w$ and the response with the lowest reward as $y_l$.

**Best-vs-random:** This policy selects the response with the highest reward as $y_w$, while $y_l$ is chosen randomly from the remaining $k - 1$ responses.

**Original annotation:** This policy chooses $y_w$ and $y_l$ from the original preference data annotated by humans or larger models.

**Rejection Sampling:** This method utilizes only the response with the highest reward as $y_w$ for each prompt $x$ and performs 1-step SFT using samples $(x, y_w)$.

**PPO:** This method dynamically generates responses $y$ for a batch of prompts and employs a reward model for their assessment. Subsequently, it maximizes the cumulative reward during RLHF training. PPO does not use any pre-generated re-

sponses.

**Proposed method:** We consider all possible combinations of $y_w$ and $y_l$ from the $k$ answers. We keep all combinations with reward gap larger than predefined threshold $\eta$.

We also investigate the performance of our proposed method under varying thresholds $\eta$, maintaining a constant temperature $\tau$. Generally, lower values of $\eta$ and $\tau$ lead to an increased size of preference data in our proposed method. In contrast to other policies that limit the sample size to the original preference data $\mathcal{D}_{\text{RM}}$ size, our proposed method considers the reward distribution per prompt and identifies more contrastive samples $(y_w, y_l)$, thereby resulting in enhanced performance. Table 1 and Table 2 summarizes the results on the MT-Bench and AlpacaEval benchmarks for Anthropic/HH-RLHF and WebGPT datasets, respectively.

In Table 1 and Table 2, our proposed method consistently demonstrates superior performance compared to other methods on the Anthropic/HH-RLHF and WebGPT datasets. All policies exhibit better performance than the SFT model, except for the best-vs-random policy and PPO on MT-Bench benchmark. This can be attributed to the best-vs-random policy's random selection of $y_l$, which, if it happens to select a high-quality response as $y_l$, can make optimization process challenging and noisy.

The best-vs-worst policy consistently outperforms other policies except our proposed method, primarily because it consistently selects high-quality pairs of contrastive samples. Furthermore, the best-vs-worst policy also outperforms the original annotation policy, despite both policies utilizing the same amount of data. This observation holds true even for pythia-6.9B-RM-WG, which is trained on the same original annotation dataset. The enhanced performance of the best-vs-worst policy can be attributed to the fact that both $y_l$ and $y_w$ are sampled from the SFT model, as opposed to utilizing responses from another language model or human annotation.

Rejection sampling method is not performing very well which can be attributed the following factors: (1) it only utilizes $y_w$ for alignment and does not take advantage of the remaining $k-1$ responses, (2) it applies 1-step SFT which can be susceptible to overfitting issues.

The performance of PPO on Anthropic/HH-RLHF surpasses that of other methods, with the exception of our proposed approach and the best-vs-worst policy. However, the performance of PPO on MT-Bench average scores declines when applied to WebGPT, primarily attributed to a low 2-turn score on MT-Bench, as detailed in Tables 4 and 5 in appendix. This can be attributed to the prompt types in the datasets, where the Anthropic/HH-RLHF dataset comprises prompts featuring multi-turn conversations between humans and AI assistants, while the WebGPT dataset exclusively involves single-turn questions. Consequently, PPO indicates an enhancement in second-turn performance on the Anthropic/HH-RLHF dataset in comparison to WebGPT within the MT-Bench benchmark.

**How does changing the threshold $\eta$ affect our performance of proposed method?** Our proposed method takes into account the reward distribution per prompt to determine pairs of $y_l$ and $y_w$ by assessing the reward gap. Lower values of $\eta$ lead to an increased generation of preference data within our proposed method because it allows selection of samples with smaller reward gaps. However, setting $\eta$ too low may lead to $y_l$ and $y_w$ being similar in quality, potentially impeding the optimization process and the convergence. In both datasets, reducing $\eta$ from 0.90 to 0.85 yields improved performance as it increases preference data generation without compromising quality. However, lowering $\eta$ further, from 0.85 to 0.80, results in a slight performance decline in two cases when using the pythia-6.9B-RM-OA reward model on MT-Bench bench. This can be attributed to a substantial increase in sample size, preventing the convergence of the optimization process and reduced quality of generated preference data.

**How does the reward model impact the results?** In our experiments, we employ two reward models with identical architectures but trained on different amount of preference data. Specifically, pythia-6.9B-RM-OA is trained on a larger preference dataset, while pythia-6.9B-RM-WG is exclusively trained on the WebGPT portion of preference datasets (detailed information is provided in section 3.2). As a result, pythia-6.9B-RM-OA exhibits superior performance in evaluating response quality in line with human preferences. Typically, a more effective reward model tends to have a higher variance in its reward distribution with longer tails, as it can differentiate the good and bad responses in a broader range. In contrast, lower quality reward models often have most rewards concentrated around the mean. Figure 2 shows the reward gap distribution for both reward models on WebGPT

| Policy | Reward Model | Sample Size | Threshold | MT-Bench (Avg score) | AlpacaEval (win %) |
|---|---|---|---|---|---|
| SFT | - | 9,000 | - | 5.12 | $60.20_{2.84}$ |
| Best-vs-worst | pythia-6.9B-RM-OA | 10,300 | - | 5.34 | $72.48_{2.59}$ |
| Best-vs-random | pythia-6.9B-RM-OA | 10,300 | - | 5.07 | $70.00_{2.64}$ |
| Original annotation | - | 10,300 | - | 5.26 | $65.33_{2.75}$ |
| Rejection Sampling | pythia-6.9B-RM-OA | 10,300 | - | 4.84 | $60.20_{2.84}$ |
| PPO | pythia-6.9B-RM-OA | 10,300 | - | 5.22 | $69.23_{2.67}$ |
| Proposed method | pythia-6.9B-RM-OA | 12,795 | 0.90 | 5.44 | $73.75_{2.54}$ |
| Proposed method | pythia-6.9B-RM-OA | 32,640 | 0.85 | **5.49** | $74.17_{2.53}$ |
| Proposed method | pythia-6.9B-RM-OA | 63,938 | 0.80 | 5.36 | $\mathbf{79.67_{2.33}}$ |

Table 1: Performance of competing methods on Anthropic/HH-RLHF dataset using different policies on MT-Bench and AlpacaEval benchmarks. A dash (-) sign indicates that the specific parameters is not needed. The SFT model is trained on Open Assistant conversation dataset. The base LLM for all experiments is Llama-2-7B. The temperature $\tau$ is set to be 1 in our proposed method. The subscript in the AlpacaEval win rate indicates the standard error.

| Policy | Reward Model | Sample Size | Threshold | MT-Bench (Avg score) | AlpacaEval (win %) |
|---|---|---|---|---|---|
| SFT | - | 9,000 | - | 5.12 | $60.20_{2.84}$ |
| Best-vs-worst | pythia-6.9B-RM-WG | 12,193 | - | 5.24 | $69.17_{2.67}$ |
| Best-vs-random | pythia-6.9B-RM-WG | 12,193 | - | 5.04 | $69.90_{2.66}$ |
| Original annotation | - | 12,193 | - | 5.14 | $65.55_{2.75}$ |
| Rejection Sampling | pythia-6.9B-RM-WG | 12,193 | - | 5.15 | $68.17_{2.69}$ |
| PPO | pythia-6.9B-RM-WG | 12,193 | - | 4.95 | $65.17_{2.75}$ |
| Proposed method | pythia-6.9B-RM-WG | 3,449 | 0.90 | 5.13 | $68.90_{2.68}$ |
| Proposed method | pythia-6.9B-RM-WG | 11,458 | 0.85 | 5.24 | $72.33_{2.59}$ |
| Proposed method | pythia-6.9B-RM-WG | 29,698 | 0.80 | 5.31 | $72.91_{2.57}$ |
| Rejection Sampling | pythia-6.9B-RM-OA | 12,193 | - | 5.23 | $71.00_{2.62}$ |
| PPO | pythia-6.9B-RM-OA | 12,193 | - | 5.11 | $69.83_{2.65}$ |
| Proposed method | pythia-6.9B-RM-OA | 12,611 | 0.90 | **5.35** | $71.91_{2.60}$ |
| Proposed method | pythia-6.9B-RM-OA | 33,755 | 0.85 | **5.35** | $\mathbf{74.92_{2.51}}$ |
| Proposed method | pythia-6.9B-RM-OA | 70,510 | 0.80 | 5.20 | $67.56_{2.71}$ |

Table 2: Performance of competing methods on WebGPT dataset using different policies on MT-Bench and AlpacaEval benchmarks. A dash (-) sign indicates that the specific parameters is not needed. The SFT model is trained on Open Assistant conversation dataset. The base LLM for all experiments is Llama-2-7B. The temperature $\tau$ is set to be 1 in our proposed method. The subscript in the AlpacaEval win rate indicates the standard error.

dataset. The red dashed line represents the threshold for preference data selection in the histograms. As depicted in Figure 2, the histogram for pythia-6.9B-RM-OA exhibits longer tails and greater variance, leading to a higher number of preference samples falling in after the dashed line. As shown in the reward model ablation study in Table 2, the pythia-6.9B-RM-OA reward model enhances the performance of our proposed method, PPO, and the rejection sampling method, underscoring the significance of a high-quality reward model. Nevertheless, the results demonstrate the robustness of our proposed method to variations in reward model quality, as it outperforms other methods even when employing the pythia-6.9B-RM-WG reward model. Additionally, results indicates that the PPO method is more sensitive to the quality of the reward model as transitioning from the pythia-6.9B-RM-OA to the pythia-6.9B-RM-WG reward model notably diminishes model performance across both benchmarks.

**How do multi-turn prompts influence performance?**

The Anthropic/HH-RLHF dataset includes prompts comprising multi-turn conversations between humans and AI assistants, while the WebGPT dataset exclusively consists of single-turn questions. Through a comparative analysis of MT-bench multi-turn scores presented in Table 4 and 5, it is evident that the incorporation of multi-turn prompts enhances the 2-turn scores for both our proposed method and PPO. Notably, our proposed method outperforms PPO. However, there is no significant impact on performance observed in the AlpacaEval benchmark as it employs only single-turn evaluation prompts. Consequently, the inclusion of multi-turn prompts in RLHF is crucial for improving the multi-turn capabilities of language models.

**How does changing the temperature $\tau$ affect**

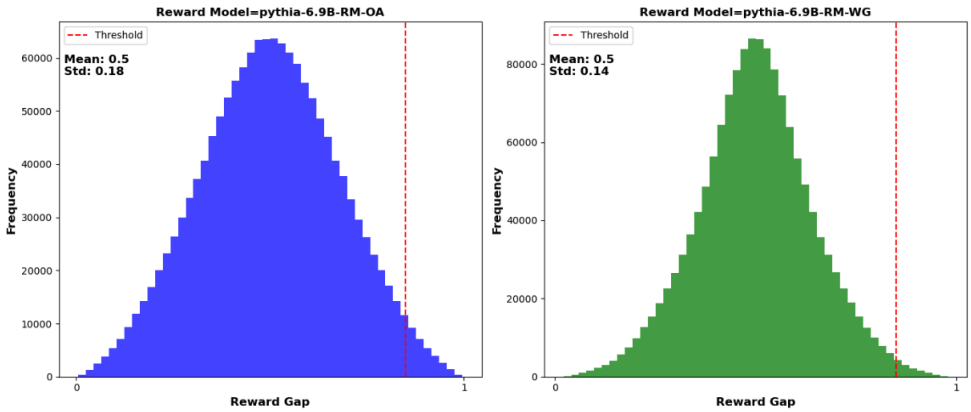


Figure 2: Histograms of reward gap for WebGPT datasets with different reward models. The red dashed line represents the threshold value of 0.85 for preference data selection. Mean and standard deviation values of reward gaps are shown in the histograms.

**our method's performance?** To analyze the impact of temperature on our proposed method, we design an ablation study where we keep the threshold $\eta = 0.85$ the same and change the value of temperature. Decreasing $\tau$ leads to a heavy-tailed reward gap distribution, generating more preference samples. Conversely, increasing $\tau$ creates a thin-tailed reward gap distribution, resulting in fewer preference samples being generated. We conduct this ablation study on Anthropic/HH-RLHF dataset and Table 3 summarizes the results. A lower temperature value increases the sample size, enhancing overall performance. As the sample size increases, a diverse variety of preference pairs emerges, encompassing both easy (with a higher reward gap) and hard (with a lower reward gap) instances. The inclusion of easy preference pairs significantly helps with the convergence of the DPO optimization process, facilitating the attainment of a superior model. In contrast, solely using hard samples may impede the optimization process, resulting in a failure to converge and yielding a policy model of inferior quality.

## 5 Discussion and Conclusion

In this paper, we proposed RS-DPO method that generates responses from the large language model directly, and leverages RS to sample synthetic preference pairs, and DPO for RLHF training. Extensive experiments show the effectiveness of RS-DPO compared to existing methods including rejection sampling (RS), proximal policy optimization (PPO) and direct preference optimization (DPO). Additionally, RS-DPO is stable, and is not as sensitive to the quality of the reward model as other methods. Our proposed method also offers a more efficient and less time-consuming solution for the alignment task as compared to PPO, minimizing resource requirements.

During RLHF training, PPO conducts online sampling from the policy model and evaluates them using the loaded reward model in real-time. Consequently, PPO necessitates loading three models during training: the initial SFT, policy model, and reward model, demanding a significant amount of GPU memory and decelerating the training process. Furthermore, the online sampling from the policy model incurs increased memory consumption as the generated sequences lengthen. In practical terms, even with 1-2 moderate GPUs, training a small-scale (e.g., 7B) LLM using PPO is unfeasible. In our experiments, we had 8 A-100 GPUs each having 40G memory, but we resorted to 8-bit quantization of both the policy and reward model to circumvent GPU memory constraints. Our proposed method conducts response sampling offline from SFT and constructs a dataset of synthetic preference data to bypass the high computational cost of PPO, while remaining viable on 1-2 moderate GPUs. Notably, the operational cost of running DPO and RS-DPO is identical; the sole disparity lies in RS-DPO performing offline SFT sampling, rendering our proposed method an on-policy reinforcement learning approach.

Moreover, as emphasized by prior researches (Singhal et al., 2023), PPO represents an unstable process prone to sensitivity towards reward model quality and hyperparameters, necessitating multiple runs to converge to a satisfactory model. For instance, in Table 2, training two models using PPO

while altering the reward model quality from high (pythia-6.9B-RM-OA) to lower (pythia-6.9B-RM-WG) significantly impacted the resulting model's quality, underscoring PPO's sensitivity. Conversely, our proposed method exhibits robustness against reward model quality, requiring only a single run to train each model successfully.

## 6 Limitations

A limitation of our work is its primary focus on the helpfulness objective derived from open-source preference datasets. Consequently, the generalizability of our findings to other objectives, such as harmlessness may be constrained. While we have demonstrated the efficacy of our proposed method on language models at 7B scale, we acknowledge that we have yet to subject our method to larger or close-source models. Despite these limitations, we maintain confidence that our proposed method demonstrates robustness towards reward model quality, and needs fewer resources compared to existing methods of RLHF training.

## References


Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

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

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2023. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

OpenAssistant. 2023. Openassistant/oasst-rm-2-pythia-6.9b-epoch-1. Accessed: 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Michael Santacroce, Yadong Lu, Han Yu, Yuanzhi Li, and Yelong Shen. 2023. Efficient rlhf: Reducing the memory usage of ppo. *arXiv preprint arXiv:2309.00754*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020a. Learning to summarize from human feedback. In *NeurIPS*.

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

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

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023a. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Ziqi Wang, Le Hou, Tianjian Lu, Yuexin Wu, Yunxuan Li, Hongkun Yu, and Heng Ji. 2023b. Enable language models to implicitly learn self-improvement from data. *arXiv preprint arXiv:2310.00898*.

Ronald J. Williams. 2004. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

## A MT-Bench Benchmark Multi-turn Results

MT-bench consists of a multi-turn question set which is deigned for testing the multi-turn conversation and instruction-following ability of LLMs. In the section, we present the MT-bench scores for all individual turns in Tables 4 and 5.

## B DPO Reward Accuracy and Reward Margin

By employing our proposed PDGRS methodology to generate preference datasets, we leverage the DPO method to fine-tune the policy model, enhancing its alignment with human preferences. Figures 3 and 4 illustrate the reward margins and accuracies achieved through DPO training across various methods on the hold-out evaluation datasets for Anthropic/HH-RLHF and WebGPT, respectively. According to the results, we observe significant correlation between increased reward margins, accuracies, and improved model performance. Our proposed preference data generation method indicates superior reward accuracy and margin in the plots, thereby underscoring the high data quality in our preference data generation approach.

## C Sample Size Controlling in RS-DPO

One advantage of our proposed method is its capacity to generate preference data by considering the reward distribution per prompt. This approach allows us to determine pairs of $y_l$ and $y_w$ by assessing the reward gap, freeing our sample size from being bound to the number of prompts in the data, unlike methods such as DPO or the Best-vs-worst method. Our results demonstrate that increasing the sample size enhances the performance of our proposed method. However, to control for the sample size's effect and showcase our method's performance when the sample size is equivalent to other methods, we subsample the generated preference data from our method to match the original number of prompts, which are 10,300 and 12,193 for the Anthropic/HH-RLHF and WebGPT datasets, respectively. Table 6 provides a summary of the results on the MT-Bench benchmark. Compared to Tables 1 and 2, our results indicate that controlling the sample size has no impact on the performance of our proposed method, which continues to outperform other methods.

## D Qualitative Examples

To conduct a qualitative comparison of model responses trained through various methods, we select sample prompts from two benchmark datasets, namely MT-Bench and AlpacaEval. Subsequently, responses are generated across all candidate models. The results of this comparative analysis are presented in Tables 7, 9 and 8.

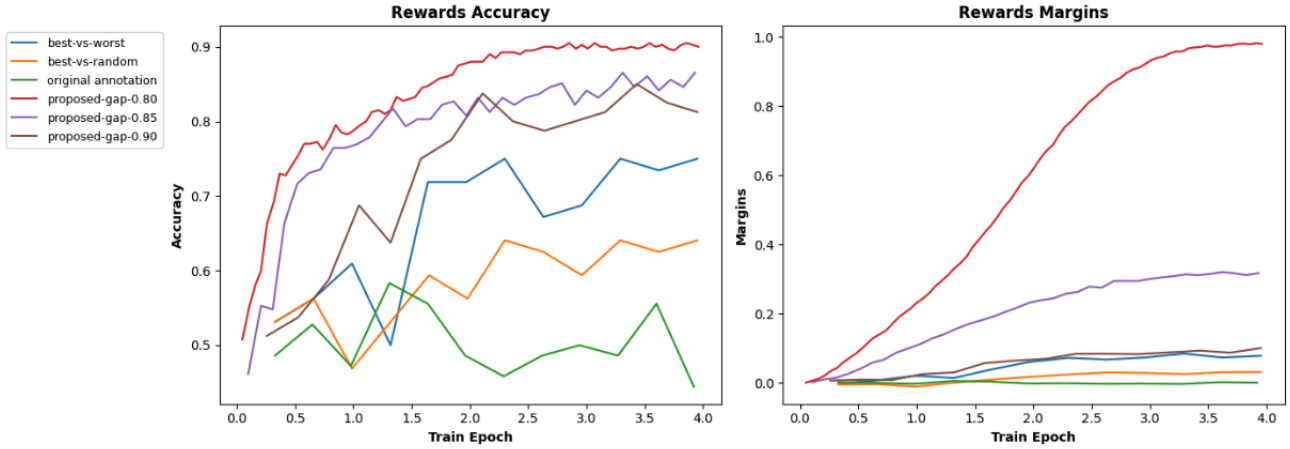


Figure 3: The left and right plots depict the reward accuracy and reward margin, respectively, of competing methods during DPO training on the Anthropic/HH-RLHF dataset.

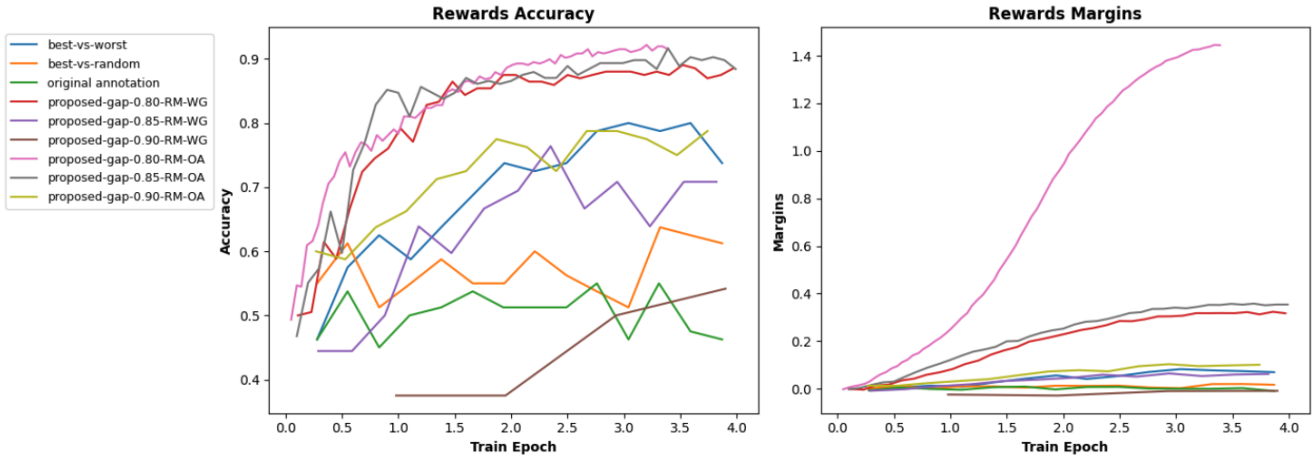


Figure 4: The left and right plots display the reward accuracy and reward margin, respectively, of competing methods during DPO training on the WebGPT dataset.

| Policy | Sample Size | Threshold | Temperature | MT-Bench (score) | AlpacaEval (win %) |
|---|---|---|---|---|---|
| Proposed method | 63,796 | 0.85 | 0.8 | 5.31 | $\mathbf{77.33}_{\mathbf{2.42}}$ |
| Proposed method | 45,668 | 0.85 | 0.9 | **5.51** | $76.92_{2.44}$ |
| Proposed method | 32,640 | 0.85 | 1 | 5.49 | $74.17_{2.53}$ |
| Proposed method | 22,951 | 0.85 | 1.1 | 5.40 | $71.00_{2.62}$ |
| Proposed method | 16,160 | 0.85 | 1.2 | 5.43 | $71.33_{2.62}$ |

Table 3: Performance of our proposed method on Anthropic/HH-RLHF dataset using different temperature $\tau$ on MT-Bench and AlpacaEval benchmarks. The base LLM for all experiments is Llama-2-7B. The reward model for all methods is pythia-6.9B-RM-OA. The subscript in the AlpacaEval win rate indicates the standard error.

| Policy | Reward Model | Sample Size | Threshold | Turn-1 | Turn-2 | Average |
|---|---|---|---|---|---|---|
| SFT | - | 9,000 | - | 5.70 | 4.54 | 5.12 |
| Best-vs-worst | pythia-6.9B-RM-OA | 10,300 | - | 6.06 | 4.61 | 5.34 |
| Best-vs-random | pythia-6.9B-RM-OA | 10,300 | - | 5.77 | 4.38 | 5.07 |
| Original annotation | - | 10,300 | - | 5.89 | 4.62 | 5.26 |
| Rejection Sampling | pythia-6.9B-RM-OA | 10,300 | - | 5.54 | 4.13 | 4.84 |
| PPO | pythia-6.9B-RM-OA | 10,300 | - | 6.03 | 4.41 | 5.22 |
| Proposed method | pythia-6.9B-RM-OA | 12,795 | 0.90 | 5.96 | **4.91** | 5.44 |
| Proposed method | pythia-6.9B-RM-OA | 32,640 | 0.85 | **6.18** | 4.81 | **5.49** |
| Proposed method | pythia-6.9B-RM-OA | 63,938 | 0.80 | 6.07 | 4.63 | 5.36 |

Table 4: Performance of competing methods on Anthropic/HH-RLHF dataset using different policies on MT-Bench benchmark. We report turn-1, turn-2, and average score from MT-Bench judged by GPT-4. A dash (-) sign indicates that the specific parameters is not needed. The SFT model is trained on Open Assistant conversation dataset. The base LLM for all experiments is Llama-2-7B.

| Policy | Reward Model | Sample Size | Threshold | Turn-1 | Turn-2 | Average |
|---|---|---|---|---|---|---|
| SFT | - | 9,000 | - | 5.70 | 4.54 | 5.12 |
| Best-vs-worst | pythia-6.9B-RM-WG | 12,193 | - | 5.85 | 4.63 | 5.24 |
| Best-vs-random | pythia-6.9B-RM-WG | 12,193 | - | 5.61 | 4.45 | 5.04 |
| Original annotation | - | 12,193 | - | 5.80 | 4.48 | 5.14 |
| Rejection Sampling | pythia-6.9B-RM-WG | 12,193 | - | 5.66 | 4.63 | 5.15 |
| PPO | pythia-6.9B-RM-WG | 12,193 | - | 5.64 | 4.26 | 4.95 |
| Proposed method | pythia-6.9B-RM-WG | 3,449 | 0.90 | 5.83 | 4.44 | 5.13 |
| Proposed method | pythia-6.9B-RM-WG | 11,458 | 0.85 | 5.86 | 4.63 | 5.24 |
| Proposed method | pythia-6.9B-RM-WG | 29,698 | 0.80 | 5.87 | 4.73 | 5.31 |
| Rejection Sampling | pythia-6.9B-RM-OA | 12,193 | - | 5.94 | 4.49 | 5.23 |
| PPO | pythia-6.9B-RM-OA | 12,193 | - | 5.94 | 4.27 | 5.11 |
| Proposed method | pythia-6.9B-RM-OA | 12,611 | 0.90 | 5.91 | **4.77** | **5.35** |
| Proposed method | pythia-6.9B-RM-OA | 33,755 | 0.85 | **6.13** | 4.55 | **5.35** |
| Proposed method | pythia-6.9B-RM-OA | 70,510 | 0.80 | 5.96 | 4.42 | 5.20 |

Table 5: Performance of competing methods on WebGPT dataset using different policies on MT-Bench benchmark. We report turn-1, turn-2, and average score from MT-Bench judged by GPT-4. A dash (-) sign indicates that the specific parameters is not needed. The SFT model is trained on Open Assistant conversation dataset. The base LLM for all experiments is Llama-2-7B.

| Policy | Reward Model | Sample Size | Threshold | Dataset | MT-Bench (Avg score) |
|---|---|---|---|---|---|
| Proposed method | pythia-6.9B-RM-OA | 10,300 | 0.90 | Anthropic/HH-RLHF | **5.37** |
| Proposed method | pythia-6.9B-RM-OA | 12,193 | 0.90 | WebGPT | **5.35** |

Table 6: Performance of our proposed method on MT-Bench benchmark with sample size controlling in Anthropic/HH-RLHF and WebGPT datasets.

| **Prompt:** You can see a beautiful red house to your left and a hypnotic greenhouse to your right, an attractive heated pink place in the front. So, where is the White House? | |
|---|---|
| Policy | Response |
| Proposed Method | The White House is the official residence and workplace of the president of the United States. It is located at 1600 Pennsylvania Avenue NW in Washington, D.C. It is a complex of buildings that includes the West Wing, the East Wing, the Residence, and the Eisenhower Executive Office Building. The White House is often referred to as "the People's House" because it is the symbol of the American people and the presidency. It is a historic landmark and a popular tourist destination. The red house, greenhouse, and heated pink place are not located near the White House, but they are likely other attractions or landmarks in the area. The White House can be easily recognized by its iconic white exterior and its prominent location in Washington, D.C. |
| SFT | The White House is the large, white building in the center of the picture. It is located in Washington, D.C., and is the official residence and workplace of the president of the United States. The red house to the left is the Blair House, which is used for official guest accommodations. The greenhouse to the right is the United States Botanic Garden, which is a research and educational institution. The attractive heated pink place in the front is the Ellipse, a park located between the White House and the Washington Monument. |
| Original Annotation | The White House is the building in the center of the picture. It is the building with the American flag flying from the roof. It is the building with the balcony and the columns. It is the building with the lawn and the trees. It is the building with the fence and the gate. It is the building with the flagpoles and the flag. It is the building with the windows and the doors. It is the building with the roof and the chimneys. It is the building with the stairs and the walkway. It is the building with the lights and the sign. It is the building with the name "The White House" written on it. It is the building that is the home of the President of the United States. |
| Rejection Sampling | The White House is the large, white building at the center of the image, directly in front of the greenhouse. The red house to the left is known as the Blair House, and it is located directly across the street from the White House. The greenhouse is the United States Department of Agriculture (USDA) Research Center, and the attractive heated pink place in the front is the USDA South Building.<br>The White House is the official residence and principal workplace of the president of the United States. It is located at 1600 Pennsylvania Avenue NW in Washington, D.C., and has been the residence of every U.S. president since John Adams in 1800. The Blair House is a historic house located at 1651 Pennsylvania Avenue NW, across the street from the White House. It serves as the official guest house for the president of the United States and is used for official entertaining and visiting dignitaries.<br>The USDA Research Center is a federal research facility that is part of the U.S. Department of Agriculture. It is located at 10301 Baltimore Avenue in College Park, Maryland, and is dedicated to research in agriculture, food, and natural resources. The USDA South Building is a federal office building that is part of the U.S. Department of Agriculture. It is located at 14th Street and Independence Avenue SW in Washington, D.C., and houses various USDA offices and programs. |
| PPO | The White House is the building in the center of the picture. It is the building with the flag flying from the roof. It is the building with the balcony and the columns. It is the building with the lawn and the trees. It is the building with the fence and the gate. It is the building with the flagpoles and the flag. It is the building with the windows and the doors. It is the building with the roof and the chimneys. It is the building with the columns and the arches. It is the building with the dome and the clock. It is the building with the flag and the stars. It is the building with the name "The White House" written on it. It is the building that is the home of the President of the United States. |

Table 7: Comparing generations obtained for a cherry-picked MT-Bench prompt from Llama-2-7B trained based on different methods.

| **Prompt:** What are the five most common causes of stress among people? | |
|---|---|
| Policy | Response |
| Proposed Method | The five most common causes of stress among people are:<br>Work: Job-related stress is one of the most common causes of stress, as it can involve long hours, tight deadlines, and demanding workloads.<br>Money: Financial stress is another common cause of stress, as it can involve worries about debt, income, and financial stability.<br>Relationships: Conflict with partners, family members, or friends can cause stress, as can the pressure to maintain healthy relationships.<br>Health: Illness, chronic pain, or concerns about one's own health or the health of a loved one can cause stress.<br>Personal problems: Events such as divorce, bereavement, or other personal difficulties can cause stress.<br>It's worth noting that these causes of stress can be interconnected, and that different people may experience stress for different reasons. It's important to take steps to manage stress and maintain good mental health. |
| SFT | The five most common causes of stress among people are:<br>1. Work or career-related issues 2. Money problems 3. Relationship difficulties 4. Health concerns 5. Personal problems<br>It is important to note that stress can also be caused by other factors such as personal beliefs, values, and experiences. Additionally, the perception of stress can vary from person to person, and what may be a source of stress for one person may not be a source of stress for another. |
| Original Annotation | The five most common causes of stress among people are:<br>1. Work or school 2. Money or financial concerns 3. Family responsibilities or relationships 4. Health concerns 5. Personal or emotional issues<br>It's important to note that the causes of stress can vary from person to person and may also be influenced by cultural and environmental factors. Additionally, it's possible for more than one cause to contribute to an individual's stress. |
| Rejection Sampling | Stress is a complex issue with many possible causes. The most common causes of stress are often related to a person's work or family life, and include: financial difficulties, work-related stress, relationship problems, and health concerns. Other common causes of stress include: personal problems, such as a difficult relationship, divorce, or the death of a loved one; and life changes, such as moving, getting married, or having a baby. Stress can also be caused by a person's environment, including: living in an unsafe neighborhood, being a victim of crime, or experiencing discrimination. |
| PPO | The five most common causes of stress among people are:<br>1. Work-related stress 2. Money problems 3. Relationship difficulties 4. Health issues 5. Personal problems<br>It is important to note that the causes of stress can vary from person to person and may be influenced by a variety of factors, including personal experiences, lifestyle, and cultural background. |

Table 8: Comparing generations obtained for a cherry-picked AlpacaEval prompt from Llama-2-7B trained based on different methods.

| **Prompt:** Why does it feel like less effort to watch 3 - hour long TV episodes back to back than a film? | |
|---|---|
| Policy | Response |
| Proposed Method | There are a few reasons why it might feel like less effort to watch several hours of TV back to back than a single film:<br>1. Scope: TV shows often have a longer scope than films, with more time to develop characters, world-building, and plot. This can make the experience of watching several hours of TV feel more immersive and less like a single event.<br>2. Pacing: The pacing of TV shows is often more episodic, with each episode having a self-contained story that builds towards a larger arc over the course of the season. This can make it easier to follow the story and feel less overwhelmed by the amount of content.<br>3. Investment: Watching several hours of TV can also create a sense of investment in the characters and story, which can make it feel less like a one-time event and more like an ongoing experience.<br>4. Comfort: Watching TV can also be a comfortable and familiar activity, with a familiar format and rhythm that can be soothing and relaxing. This can make it easier to engage with the content for extended periods of time.<br>However, it's important to note that both films and TV shows can be enjoyable and engaging experiences, and the format preference is a personal one that depends on individual preferences and circumstances. Some people might prefer the immersive experience of a film, while others might find the longer format of TV more conducive to their lifestyle or entertainment preferences. Ultimately, the choice between films and TV shows is a matter of personal preference and can vary depending on the specific content, genre, and format preferences of the individual. |
| Original Annotation | There are several reasons why it may feel like less effort to watch three-hour long TV episodes back to back than a film. Firstly, the shorter duration of a film may make it seem more daunting to commit to watching it in one sitting, while the longer duration of a TV episode may feel more manageable. Additionally, the episodic format of TV shows often allows for natural breaks and pacing, which can make it easier to stay engaged and avoid feeling overwhelmed.<br>Another reason may be the structure of the content itself. TV shows tend to have more character development, plot twists, and subplots than films, which can make them feel more complex and engaging. On the other hand, films often have a more straightforward narrative structure and focus on a single story arc, which can make them feel more straightforward and easier to follow.<br>Finally, the cultural context in which we consume media may also play a role. With the rise of binge-watching and streaming services, it has become more common for people to watch multiple episodes of a TV show in one sitting. This has created a cultural expectation that may make it feel more normal and acceptable to watch longer periods of time, while the traditional film-going experience may still feel like a more significant investment of time and effort.<br>In summary, the combination of shorter duration, episodic structure, complex content, and cultural expectations may all contribute to the feeling that it is less effort to watch multiple TV episodes back to back than a film. However, it's important to note that this is a subjective experience and everyone's preferences and priorities may differ. |
| PPO | There are several possible reasons why it might feel like less effort to watch three-hour long TV episodes back to back than a film. One reason could be that the longer format of the TV episodes allows for more breaks and pauses, which can make the viewing experience feel less taxing. Additionally, the pacing of TV episodes can be more leisurely, which can make the content feel less overwhelming and easier to digest. Finally, the longer time commitment required for a TV series might make it feel like a more manageable task than a two-hour film, which can feel like a significant investment of time and attention. Ultimately, the perception of effort involved in watching content may be influenced by a variety of factors beyond the inherent length or format of the content itself. |

Table 9: Comparing generations obtained for a cherry-picked AlpacaEval prompt from Llama-2-7B trained based on different methods.