ChatGLM-RLHF: Practices of Aligning Large Language Models with Human Feedback

Zhenyu Hou[1,2*] **Yilin Niu**[1*] **Zhengxiao Du**[1,2] **Xiaohan Zhang**[1] **Xiao Liu**[1,2]
Aohan Zeng[1,2] **Qinkai Zheng**[1,2] **Minlie Huang**[2] **Hongning Wang**[2]
Jie Tang[2] **Yuxiao Dong**[2]
[1]Zhipu AI, [2]Tsinghua University

Abstract

ChatGLM is a free-to-use AI service powered by the ChatGLM family of large language models (LLMs). In this paper, we present the ChatGLM-RLHF pipeline—a reinforcement learning from human feedback (RLHF) system—designed to enhance ChatGLM's alignment with human preferences. ChatGLM-RLHF encompasses three major components: the collection of human preference data, the training of the reward model, and the optimization of policies. Throughout the process of integrating ChatGLM-RLHF into production, we encountered and addressed several unprecedented challenges. We introduce the strategies to mitigate reward variance for stabilized large-scale training, implement model parallelism with fused gradient-descent, and design regularization constraints to avoid catastrophic forgetting in LLMs. Experiments show that ChatGLM-RLHF brings significant improvements in alignment tasks compared to the supervised fine-tuned (SFT) version of ChatGLM. For instance, it achieves on average 15% more wins against ChatGLM-SFT in Chinese alignment tasks. The work presents our practices of aligning LLMs with human preferences, offering insights into the challenges and solutions in RLHF implementations.

1 Introduction

Large language models (LLMs) Brown et al. (2020); Zhang et al. (2022); Workshop et al. (2022); Zeng et al. (2023); Touvron et al. (2023b) have remarkably advanced the capabilities of machines in language understanding and generation. The most notable example is ChatGPT Achiam et al. (2023), which demonstrates strong capability in responding to users' queries and following their instructions. Different from the pre-trained GPT-3 model Brown et al. (2020), ChatGPT (GPT-3.5, GPT-4 Achiam et al. (2023), and beyond) are further post-trained, with supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) Ouyang et al. (2022), to better align with human preferences. We have developed ChatGLM —a free-to-use AI service powered by the ChatGLM family of LLMs Zeng et al. (2023). Similar to ChatGPT, ChatGLM was pre-trained for trillions of multilingual tokens and post-trained with both SFT and RLHF.

The goal of RLHF is to encourage LLMs to better align with human preferences and values, i.e., to produce more helpful, more accurate, and safer responses Bai et al. (2022b); Achiam et al. (2023); Ouyang et al. (2022). Specifically, the RLHF techniques typically treat human preferences as rewards and employ reinforcement learning algorithms, such as Proximal Policy Optimization (PPO) Schulman et al. (2017), to realize the goal of alignment. Despite the significant community

---
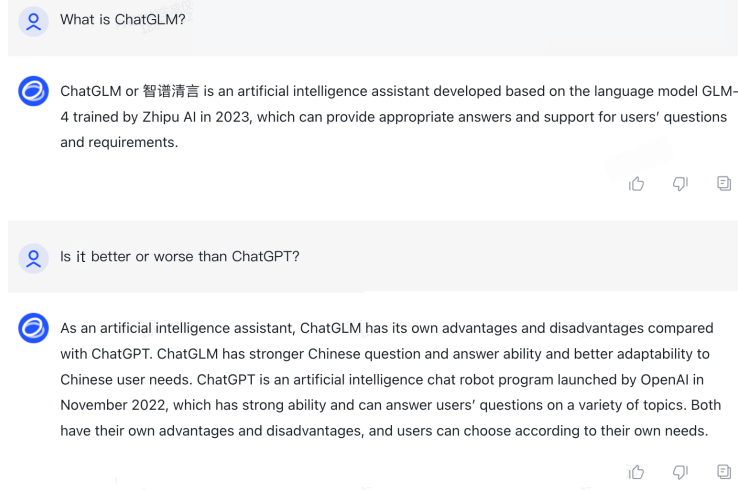
[*]ZH and YN contributed equally.

Figure 1: An illustrative example of ChatGLM (`chatglm.cn` accessed on Feb. 8, 2024).

efforts Touvron et al. (2023b); Bai et al. (2023); Dai et al. (2023); Sun et al. (2023) in applying RLHF to perform LLMs alignment, since the inception of ChatGPT, our experience in constructing a practical RLHF training framework for ChatGLM has revealed unknown technical challenges.

First, the collection and modeling of reliable human preferences is critical to the effective use of human feedback. However, biased Cui et al. (2023); Longpre et al. (2023) and deceptive preferences Bai et al. (2022a) in human annotations are often inevitable. Second, the reward model is readily impacted by biased preferences, which can lead to the learning of shortcut features. This issue may drastically compromise the precision and generalization of the reward model. Third, a scalable and robust training framework is required to ensure efficient and effective optimization of LLMs across different scales, as well as to cover all potential corner-cases to meet the requirements for online deployment. For instance, applying PPO on LLMs of 32B parameters requires at least 8 or 16 A100 (8×80G) servers for 2-3 days, respectively. However, most existing efforts focus on parameter-efficient tuning Sun et al. (2023); Yao et al. (2023) or small-scale models Zheng et al. (2023b); Wu et al. (2023).

In this paper, we present the ChatGLM-RLHF pipeline, which has been used to improve ChatGLM's alignment with human preferences, with a particular focus on the Chinese language. This pipeline serves as a critical component in improving ChatGLM's performance to generate responses that more accurately reflect human values and expectations. We detail our approach for designing and developing the pipeline—human preference annotation, reward model training, policy optimization, as well as practices of addressing the challenges associated with these steps.

First, we set up a routine system to *collect human preference annotations*. To ensure the consistency among annotators, we employ a pairwise comparison mechanism. Specifically, each annotator is asked to select the preferred response from two outputs generated by our SFT model, guided by annotation guidelines that cover three key aspects of helpfulness, harmlessness, and fluency. We also develop a post-filtering process to remove undesirable annotations, such as cyclic and tie preferences. Second, we train a *reward model* on the collected preference dataset as a proxy for the responses an average human user would favor. To train a reliable proxy, we develop strategies to prevent the reward model from taking shortcuts or learning unexpected biases, such as a biased preference towards longer yet not really helpful outputs Singhal et al. (2023); Shen et al. (2023). Finally, by using the reward model as the proxy for human preferences, we apply an online RL algorithm *PPO* and an offline RL algorithm *DPO* Rafailov et al. (2023) to align the language model.

Best Practices. To support scalable RLHF training, we implement a robust training framework with important practical designs, including the strategies of reward variance reduction to stabilize large-scale training, model parallelism with fused gradient-descent, and regularization constraint to avoid catastrophic forgetting. In particular, we summarize the following lessons learned over the course of developing ChatGLM-RLHF:

- Establishing criteria and detailed reference dimensions for annotation contributes to more reliable and consistent human preference.

- Eliminating bias from the reward model can serve as an efficient and powerful approach to more accurately reflect genuine human preferences and reduce the influence of spurious correlation.

- Training stability can be substantially improved by subtracting a baseline reward from the original reward during PPO training.

- Incorporating next-token-prediction loss of SFT data can reduce capability shifting in RLHF training.

Extensive experiments on ChatGLM-6B and ChatGLM-32B demonstrate that ChatGLM-RLHF can significantly improve the performance of ChatGLM, enabling it to produce more helpful, safe, and aligned responses. The ChatGLM models, refined through the ChatGLM-RLHF pipeline, are supporting the online services available at *chatglm.cn* and mobile applications on iOS and Android platforms.
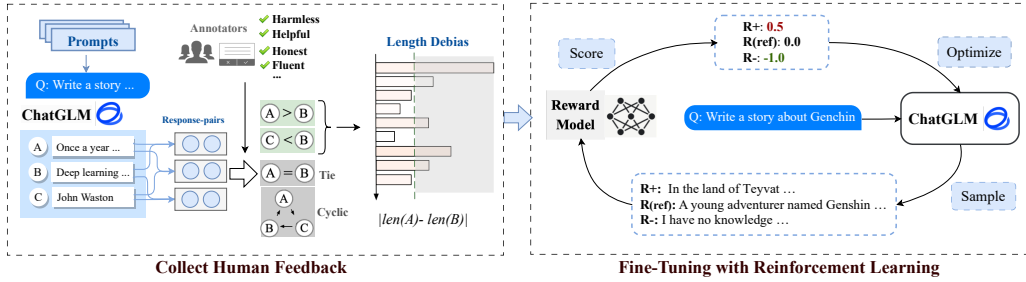
2 Related Work

Figure 2: The overall design of the ChatGLM-RLHF pipeline. We first set up a comprehensive system to collect human preferences over responses of ChatGLM and remove unexpected patterns and potential biases in the data. Then we train a reward model to predict human preferences and employ reinforcement learning to optimize ChatGLM to generate responses assigned with higer rewards.

Large Language Models. Large Language Models (LLMs), particularly those employing self-supervised learning techniques, have become a focal point in the field of Natural Language Processing (NLP) due to their ability to encode a vast array of knowledge within extensive parameter sets. This attribute contributes to their remarkable capabilities across a wide range of tasks. Prominent examples of such models include GPT-3 Brown et al. (2020), PaLM Chowdhery et al. (2023), OPT Zhang et al. (2022), BLOOM Workshop et al. (2022), GLM-130B Du et al. (2021); Zeng et al. (2023) and LlaMA Touvron et al. (2023a,b).

LLMs can be prompted to perform a range of NLP tasks. They can effectively leverage given examples in the context Brown et al. (2020) to solve upcoming tasks yet without explicit finetuning. This in-context learning has primarily demonstrated LLM's ability to solve diverse and challenging tasks yet still requires elaborated few-shot exemplars as activation. However, these models often express unintended behaviors such as making up facts, generating biased or toxic text, or simply not following user instructions Bommasani et al. (2021); Kenton et al. (2021); Bender et al. (2021). To enable LLMs to respond to human instructions in zero-shot setting, instruction tuning Wei et al. (2022); Chung et al. (2022) trains LLMs with natural language instructions consisting of human-written instructions and responses. Such targeted fine-tuning substantially augments the LLMs' ability to generalize to diverse tasks beyond the training dataset and significantly boosts the performance Longpre et al. (2023). Recently, Supervised Fine-Tuning (SFT) Ouyang et al. (2022), which could be regarded as an extension of instruction tuning, is proposed to further enhance the alignment of LLM outputs with human expectations using high-quality and diverse human-written datasets. The quality of instruction data has been proved to be significantly critical Zhou et al. (2023);

Touvron et al. (2023b) in alignment. This approach activates and bolsters the LLMs' capabilities in various domains, including conversation, reasoning, and writing, among others.

Aligning LLMs with Human Feedback. LLMs as AI assistants are expected to follow human instructions and be helpful, honest, and harmless to align with human values and preferences Ouyang et al. (2022); Bai et al. (2022b). Despite the effectiveness of instruction tuning and SFT, gathering accurate and expert annotations is costly and becoming more difficult with the increasingly stronger ability of LLMs, as higher quality data and stronger supervision are a must to direct model training in SFT. In comparison, collecting human judgments on response quality is more feasible because it is easier for human to serve as a judge than to provide the gold answer, as demonstrated in discriminative-generative gap Zheng et al. (2023c). Plenty of studies Touvron et al. (2023b); Ouyang et al. (2022); Bai et al. (2022b,a); Ganguli et al. (2023) have focused on refining LLMs using human preferences and found notable improvements in areas such as summarization and instruction-following. InstructGPT Ouyang et al. (2022) first trains a reward model to fit human preference and then optimizes LLMs to maximize rewards through reinforcement learning, i.e., Proximal Policy Optimization Schulman et al. (2017), which becomes standard RLHF paradigm for following works Touvron et al. (2023b); Bai et al. (2022b). Another line of works Rafailov et al. (2023); Zhao et al. (2023); Song et al. (2023) is to directly optimize the policy model with human preference without training a reward model. DPO Rafailov et al. (2023) derives the policy objective from Bradley-Terry Bradley and Terry (1952) reward preference. SLiC Zhao et al. (2023) directly contrasts the positive and negative samples with a margin loss. PRO Song et al. (2023) extends pairwise to listwise data and utilizes multiple negative samples. Compared to PPO, these works behave similarly to off-policy reinforcement learning which could be more efficient yet might sacrifice performance.

Despite open-source practices Yao et al. (2023) and studies Sun et al. (2023); Zheng et al. (2023b) on PPO optimization, almost all of them are conducted on small-scale models or parameter-efficient tuning with well-prepared academic benchmarks. In this work, we set up the RLHF framework from scratch and optimize the ChatGLM model, taking into account a range of practical constraints while placing emphasis on scalability.

3 ChatGLM-RLHF

We present the design of ChatGLM-RLHF, including human preference data collection, reward model training, and policy model optimization. Uniquely, we introduce challenges encountered and practical solutions proposed for large-scale RLHF model training.

3.1 Learning from Human Preferences

Reinforcement Learning from Human Feedback (RLHF) for large language models (LLMs) typically initiates by fine-tuning a pre-trained language model with high-quality human-written data in a supervised finetuning (SFT) manner. This step yields a preliminary aligned model π_0, called the SFT model, which can respond to human instructions yet may not behave helpfully and safely. To further align the LM with human values and preferences, RLHF acts upon the SFT model. It views human feedback (or preference) for model-generated texts as reward for maximization, which is different from training through next-token-prediction. Following the common setting in reinforcement learning, RLHF first trains a reward model $r_\phi(x, y)$ as a proxy to model human preferences. Then the reward model is utilized to optimize policy model $\pi_\theta(y|x)$ to generate responses that are rewarded higher but are not drifting too far from the SFT model, where (x, y) represents prompt and response pair and $\{\phi, \theta\}$ are learnable parameters.

Subsequent parts of this section will delve into the details of the RLHF design in ChatGLM, including gathering human preference data, modeling human preferences, and fine-tuning the policy model to better align with human values.

3.2 Data Collection and Processing

Prompt Collection To cover the diversity of human intents and preferences, it is important to collect a comprehensive prompt set for later response generation and comparison. We collect prompts based on the demands of real-world application scenarios and apply quality filtering to select high-quality

Table 1: Examples of prompts with varied quality.

Clear intention; clear semantics; answerable:
Assist me in crafting a three-day travel itinerary to Beijing with a budget of under 5000.
Unclear intention:
The gentleman attended the meeting, dressed in formal attire.
Incomprehensible semantics:
Christmas, Reindeer, Christmas Tree
Unanswerable prompt:
What is the winning lottery number for tomorrow?

prompts. Specifically, we establish a taxonomy for the possible intentions of prompts and ensure that each category of intentions is accompanied by enough training prompts. After that, a quality classifier is employed to score each prompt in three aspects:

- The *intention* of the prompt is clear, ambiguous, or totally unclear.
- The *semantic* is clear, guessable, or totally incomprehensible.
- The prompt is *answerable* or beyond models' capabilities, such as predicting lottery numbers.

The answerable prompts with clear intentions and semantics are utilized for training reward/policy models (see Table 1 for examples). Besides, we also include several rule-based filters to select informative prompts, such as filtering out too short prompts.

Preference Annotation Reward model is trained with human feedback on the model-generated responses. To collect human preference, each annotator is provided with one prompt and two responses and asked to determine which response is preferred. They are encouraged to make a decision based on helpfulness and safety:

- *Helpfulness* encompasses multiple aspects including the extent to which a response fulfills every requirement of the prompt, provides accurate and valuable information, and maintains logical consistency. Additionally, we posit that responses lacking linguistic fluency are disqualified from meeting the requirements of users. Therefore, instances of grammatical errors or unexpected language mixing, such as a mixture of Chinese and English, can detrimentally impact the perceived *helpfulness* of a response.
- The focus of *safety* is to ascertain whether responses contain harmful or toxic content, as well as content that could potentially provoke controversy.

Annotators, based on their assessments of helpfulness and safety, will decide *preference* from the perspective of overall quality. This evaluation, while inherently subjective, is crucial in the context of alignment. As the reward model is expected to assess diverse responses, various chat models and sampling strategies are utilized to generate diverse responses for annotation. Table 2 illustrates the statistics of the preference data used for the reward model training.

3.3 Reward Model Training

Reward model is designed to assess the quality of responses, which serves as the training signal for policy improvement. As annotators are asked to take helpfulness and safety into account during the annotation of preference, preference can thus reflect the overall quality of the responses in these aspects. Consequently, we exclusively utilize preference annotations for our reward model training.

In practice, the reward model is initialized with ChatGLM-SFT, and then trained on the preference data under the loss function

$$\mathcal{L}_{RM} = -E_{(x,y_w,y_l)\sim\mathcal{D}_{RM}}[\log(\sigma(r_\phi(x,y_w) - r_\phi(x,y_l)))]. \tag{1}$$

where x denotes the prompt, y_w is the preferred response to y_l by the annotators. The reward model r_ϕ assigns a scalar value to each *(prompt, response)* pair.

Length Bias Reduction There exist some inherent biases within the preference data, such as the inclination of humans to prefer longer, well-structured responses. Such biases may mislead the

Table 2: Stat. of human preference data for reward training.

Num. of Comparisons	Avg. # Turns per Dialogue	Avg. # Tokens in History	Avg. # Tokens in Prompt	Avg. # Tokens in Response
221,866	2.4	314.1	104.1	267.7

reward model to over-emphasize these secondary features, thereby overlooking the content quality of responses. To alleviate the influence of length bias on the reward model, we devise a debiasing method named 'Bucket-Based Length Balancing'.

The first step in our method involves calculating the length difference between the two responses in each preference pair, i.e. $d = abs(|\text{tokenize}(y_w)| - |\text{tokenize}(y_l)|)$. Following this, assign the preference pairs with similar length differences into the same buckets, each of which corresponds to a specific difference range. In the final step, within each bucket, balance the number of examples where the better/worse response is longer.

Note that the efficacy of this method in mitigating length bias is determined by the granularity of buckets. Finer ranges of buckets result in more effective length bias reduction.

Stable Training During the training of reward model, we observe substantial volatility in the distribution of scores predicted by the reward model, which suggests unstable training. To alleviate this problem, we introduce a new loss component

$$\mathcal{L}_{REG} = r_\phi(x, y_w)^2 + r_\phi(x, y_l)^2, \tag{2}$$

which resembles L2 regularization loss. This loss term imposes a Gaussian prior with a mean of zero on the score distribution, thereby constraining the volatility of the score distribution.

3.4 Policy Model Training

Setting up Training Data In this stage, the aim is to leverage the reward model r_ϕ to guide the optimization of the policy model π_θ to elicit responses of increased reward value. The model is expected to receive both positive and negative feedback and is capable of discerning disparate responses of varied quality, thereby progressively augmenting its response generation in line with human preferences. To realize this, it is crucial for the model to generate responses spanning a spectrum of quality and thus the model can learn and improve from the reward feedback. Consequently, prompts leading to almost the same responses must be systematically excluded to ensure the effectiveness of this process.

More specifically, we conduct a preprocessing step to filter out less valuable data by means of the reward model. For each prompt, the model generates K responses. If the variance in rewards for these responses is smaller than a margin ϵ, it indicates that the prompt offers limited opportunity for exploration and improvement against the current model, and is therefore removed from the training dataset. Consequently, the final training dataset includes only those instances where significant variation and diversity are observed in the responses generated by the model.

Proximal Policy Optimization (PPO) is an online reinforcement learning framework for policy improvement. During PPO, we seek to optimize the policy model by maximizing the cumulative reward.

$$\text{argmax}_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y_g \sim \pi_\theta}[r_\phi(x, y_g)] \tag{3}$$

where π_θ is the policy model and θ is its learnable weights. In each training iteration, the policy model first generates responses $\{y_i\}$ for a set of prompt $\{x_i\}$. Then the reward model evaluates and assigns scores to each response, leading to $\{r(x_i, y_i)\}$, which is used to direct the gradient descent update of the policy model.

Since the reward reflects approximate human preference and could be inaccurate, to avoid the policy diverging too far from the initial SFT model π_0, a penalty term based on KL divergence is usually added as part of the reward as regularization and also helps maintain training stability.

$$\begin{aligned} r(x, y_g) &= r_\phi(x, y_g) - \beta D_{KL}(\pi_\theta(y_g|x)||\pi_0(y_g|x)) \\ &= r_\phi(x, y_g) - \beta \log \frac{\pi_\theta(y_g|x)}{\pi_0(y_g|x)} \end{aligned} \tag{4}$$

Reward Bias Reduction In the previous section, we make efforts to mitigate potential biases in reward modeling, such as length bias. Despite these efforts, it is observed that biases could not be entirely eliminated. These biases predominantly stem from two factors: 1) Value instability. The reward model utilizes a pairwise training loss; however, it does not impose explicit constraints regarding the numerical values assigned to responses in relation to their quality. Consequently, this lack of definitive guidance necessitates the implementation of reward or advantage normalization in the preliminary stages of PPO training to maintain stability. 2) Task/sample bias. The objective to enhance the general capabilities of the policy model necessitates training data that encompasses a wide range of tasks and prompts, including creative writing, mathematical reasoning, and role-playing. This diversity leads to the reward model assigning highly variable rewards across different prompts and tasks. This variability is attributed to differences in response style, length, and the Discriminativeness of the reward model, among other factors. Figure 3 illustrates the significant disparities in reward scores both within a single task and across different tasks. Consequently, a response that receives a high reward cannot be reliably deemed as high-quality without considering the influence of its associated prompt and task.

To overcome the above drawbacks, we introduce a reference baseline to avoid the variability from absolute value. The intuition is that the target is to enable the policy model to generate a better response than the reference one as much as possible, no matter its absolute reward. This corresponds to the initial objective of human preference and reward model training which excels at pairwise comparison rather than pointwise scoring. To be specific, given a prompt x, we generate a response y_{ref} during pre-processing as the reference response. During policy training, we regard the reward difference between the generated response y and the reference response y_{ref} as the optimization target,
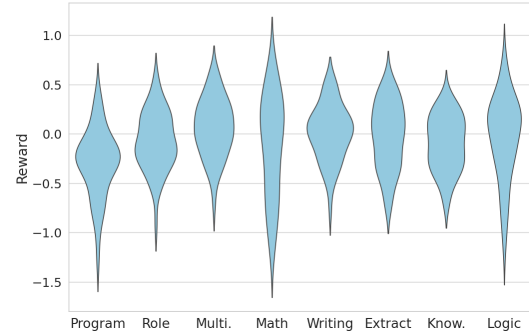

Figure 3: Reward distributions on different tasks.

$$r(x, y) = (r_\phi(x, y_g) - r_\phi(x, y_{ref})) - \beta D_{KL}\left(\pi_\theta(y_g|x)||\pi_0(y|x)\right) \tag{5}$$

At the warm-up training stage, y and y_{ref} are drawn from almost the same distribution, and thus $r(x, y)$ starts from near zero, which also helps stabilize the training even without other normalization tricks. Wu et al. (2023) adopts a similar idea to replace absolute feedback with relative feedback

Capability Forgetting The RLHF is expected to increase the response quality and better align them with human preferences. However, as a post-training step subsequent to SFT, we also observed unexpected behaviors in the policy after the RLHF stage. The model shows a reduced capability in handling specific scenarios, including wrong self-identification and invalid JSON format, and shifts a little far from the original model unexpectedly. For example, the model is trained to claim itself as "ChatGLM" after SFT yet it would describe itself as "an AI assistant" after RLHF if asked *"who are you?"*. This behavior could be attributed to the problem of difference among data distributions or the inability of the reward model in such nuanced details. The problem of capability forgetting (or shifting) gets little attention in previous studies. But it is critical for online deployment, especially for addressing diverse edge cases in real-world scenarios. For example, output in JSON format is crucial for tasks involving semantic analysis or answer extraction in practical applications. The emergence of capability forgetting can be attributed to the variance in data distribution between SFT and RLHF training, which encapsulates different facets of a model's capabilities.

To overcome the issue of capability forgetting, we propose to incorporate an extra supervised next-token-prediction loss as anadditionaln regularization besides the KL divergence, when performing reward maximization. This is intended to preserve the pre-existing abilities of the SFT model. The high-level idea is to encourage the model's outputs to align with human preferences through RLHF and leveraging next-token prediction to capture more granular signals. To be more specific, in each training step, the training involves two aspects of data and objective: 1) RLHF training with prompts and model-generated responses for preference optimization, $\mathcal{L}_r = -\sum_{x \sim \mathcal{D}} r(x, y_g)$; 2) next-token-prediction with a small amount of human-annotated (prompt, response) pairs $\mathcal{D}_S$, which

are specific to particular tasks and serve as supervised regularizations,

$$\mathcal{L} = \mathcal{L}_r - \beta \cdot \sum_{(x,y)\sim\mathcal{D}_S} \log \pi_\theta(y|x) \tag{6}$$

A simplified alternative: Direct Preference Optimization (DPO) PPO has shown special advantages as an online algorithm, but it also brings additional technical difficulties in efficient parallel training in large language models as it involves both gradient decent update and response sampling during training. Direct Preference Optimization (DPO) Rafailov et al. (2023) has been proven to be a simple yet effective alternative to PPO. DPO proposes to directly learn from annotated preference data without training a reward model. In our practice, we also implement DPO for comparison and find it is more flexible yet very effective. Different from the implementation reported in the original paper, we still train a reward model and regard it as a discriminator to help construct training data pairs with varied reward scores(x, y_w, y_l). This way separates training data construction and data annotation and is thus more scalable. In DPO, the policy is trained to optimize the following objective:

$$\mathcal{L}_r = - \sum_{(x,y_w,y_l)\sim\mathcal{D}} \log \sigma(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_0(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_0(y_l|x)}) \tag{7}$$

Parallel Training To efficiently train large models, the policy is trained with a combination of data parallelism and model parallelism Rasley et al. (2020); Narayanan et al. (2021). Model parallelism is further divided into tensor parallelism, which involves the division of tensors, and pipeline parallelism, which distributes different layers of a network across multiple GPUs. Pipeline parallel can effectively reduce the communication overhead between GPUs during training with large batch size. Typically, both the pretraining phase and supervised finetuning leverage a three-dimensional parallelism strategy—combining data, tensor, and pipeline parallelism—to optimize efficiency. In the context of DPO, which relies solely on pre-generated responses, its training aligns with that of SFT, allowing for the application of three-dimensional parallelism. But things are different for PPO, which involves the fusion of both gradient descent and inference within its training process. Each generation comprises hundreds or thousands of forward operations and would incur a number of unexpected bubbles in the pipeline parallel and significantly slow down the generation. The generation generally accounts for more than 80% training time, and it is crucial to avoid these bubbles for efficiency. Consequently, in PPO training, we adopt a two-dimensional parallelism approach, combining data and tensor parallelism, to circumvent these bottlenecks. There are public implementations based on fully-sharded data parallel Yao et al. (2023); Havrilla et al. (2023) but their efficiency is not satisfactory when finetuning full parameters of large-scale models due to the heavy communication cost.

RLHF v.s. Rejection Sampling Previous studies Touvron et al. (2023b) have shown the effectiveness of Rejection-Sampling Fine-Tuning (RFT) as a post-SFT step in enhancing the performance of model alignment. RFT involves generating multiple outputs from the policy model and selecting the best candidates with the reward as the gold standard for further finetuning. The main difference between RFT and DPO is that DPO utilizes both positive and negative pairs for optimization, whereas RFT focuses on the positive instances. Both methods are constrained to pre-generated responses for training. PPO, an online algorithm, dynamically updates the model weights based on responses generated from the latest policy during training, which would generally produce better results.

4 Experiments

4.1 Experimental Setup

We conduct experiments based on ChatGLM Zeng et al. (2023); Du et al. (2021) and test the effectiveness of DPO and PPO as the post-training step after SFT, as RLHF is always implemented based on an SFT model. We mainly focus on the improvement of the RLHF-aligned model over SFT model via both automatic evaluation and human evaluation.

Training Details For PPO training, the policy and critics models share the same learning rate 1e-6 and are trained for around 600 to 800 iterations. Neither reward nor advantage normalization is adopted as reference rewards already lead to a stable training process in our experiments. At inference

Table 3: Results of automatic evaluation on AlignBench (Chinese alignment) rated by gpt-4-0613. *Prof. Know.*: professional knowledge. *Fund. Lang.*: fundamental language. *Logic.*: logical reasoning.

	Prof. Know.	Advanced Chinese	Fund. Lang.	Math	Writing	OpenQA	Role Play	Logic.	Overall
GPT-3.5-turbo-0613[1]	6.77	5.81	6.71	5.68	7.03	7.29	7.28	5.02	6.08
GPT-4-0613[1]	7.94	6.93	7.81	7.56	7.93	7.42	7.51	7.37	7.53
GPT-4-1106-preview[1]	8.65	7.33	7.99	7.80	8.67	8.61	8.47	7.66	8.01
GLM-4 (0116)	8.52	8.19	7.29	7.26	8.27	8.50	8.53	7.30	7.75
Chinese-LLaMa-2-7B-Chat[1]	4.13	4.26	4.31	2.29	4.63	4.50	4.91	3.07	3.57
Qwen-7B-Chat[1]	5.66	5.74	6.40	3.62	6.31	6.26	6.19	3.83	4.91
ChatGLM3-6B-SFT	4.80	5.17	5.21	2.71	6.72	6.55	6.34	4.02	4.58
ChatGLM3-6B-DPO	5.64	4.60	4.94	3.41	6.85	7.29	6.66	3.81	4.80
ChatGLM3-6B-PPO	5.93	5.08	5.28	3.57	7.13	6.84	6.16	3.90	4.90
DeepSeek-67B-Chat	7.37	6.52	7.12	5.71	7.20	7.58	6.91	5.79	6.43
DeepSeek-67B-Chat-DPO	7.71	7.47	7.29	6.13	7.51	7.82	7.83	5.41	6.69
ChatGLM-32B-SFT[2]	7.47	6.62	6.53	5.87	7.45	7.13	7.38	5.42	6.37
ChatGLM-32B-DPO	7.27	6.41	6.87	6.12	7.97	8.34	8.17	5.60	6.68
ChatGLM-32B-PPO	7.27	6.97	7.06	5.85	7.92	8.18	7.96	6.02	6.75

1 The results are from the AlignBench paper, which employs varying generation temperatures for different tasks. Notably, all ChatGLM models produce responses in a greedy setting. In our experiments, the performance in the greedy setting is typically slightly lower than in the original setting but is stable.
2 The version of ChatGLM is ChatGLM-32B-2310.

time, the policy top-p is set to 0.9 to generate diverse responses. In this section, we report the results on ChatGLM-6B and ChatGLM-32B models.

4.2 Automatic Evaluation

Setup To evaluate the alignment performance on tasks in Chinese, we assessed the ability of our trained language models across various fields using the AlignBench test set Liu et al. (2023), a benchmark for open-ended questions. AlignBench comprises 8 main categories and 36 subcategories of tasks, covering a total of 683 questions. Alongside each question, AlignBench provides a reference answer and evaluation criteria, facilitating the automatic assessment of response quality by LLM-as-judge. We follow the original setting and use GPT-4 as the judge. The evaluation framework assembles the evaluation principles, the question, the model-generated response, and a reference answer as the input prompt to the LLM judge. The LLM-judge then performs the intended analytical evaluation and assigns a score ranging from 1 to 10 to the response.

We use the official code repository of AlignBench, available on GitHub[1]. We generated responses to all questions with a greedy strategy to ensure the stability of the results and reduce variance, which is a little different from the original setting that sets different temperatures for different tasks.

Results The overall results are illustrated in Table 3. We compare with ChatGPT Achiam et al. (2023) and various open-source models Bi et al. (2024); Bai et al. (2023); Touvron et al. (2023b). The results of the ChatGLM-32B show that RLHF methods, including both DPO and PPO, substantially improve the SFT model across a wide range of tasks in Alignbench. Notably, tasks such as Writing, OpenQA, and Role-Play exhibit more significant improvements. This suggests that the reward model is particularly suited to tasks related to creative writing, whereas it demonstrates limitations in tasks requiring advanced reasoning abilities, such as math and logic. This observation aligns with the intuitive understanding that human preferences tend to lean more toward stylistic and formatting elements rather than deep critical thinking skills. Furthermore, PPO demonstrates a slight edge over DPO, with an average improvement margin of approximately 0.07. This outcome aligns with expectations, considering that PPO's more intricate design demands significantly greater resources during training compared to DPO.

---

[1]`https://github.com/THUDM/AlignBench`

Table 4: Results of human evaluation of ChatGLM-32B on internal test set. We report the proportion of wins and ties of the PPO and SFT model on different tasks.

	PPO win	Tie	SFT win	Δ(PPO-SFT)
Multilingual	0.32	0.44	0.24	0.08
Creative Writing	0.40	0.52	0.08	0.32
Role-play	0.42	0.42	0.16	0.26
Knowledge	0.36	0.50	0.14	0.22
Semantic Extraction	0.30	0.50	0.20	0.10
Math.	0.16	0.70	0.14	0.02
Programming	0.22	0.72	0.06	0.16
Logic	0.20	0.64	0.16	0.04
Overall	0.30	0.55	0.15	0.15

4.3 Human Evaluation

Setup. In addition to automatic evaluations, we also incorporate human evaluations to assess the effectiveness of RLHF. Human evaluation is important because they're less likely to be biased as long as the rules are clearly defined. We collect a diverse human evaluation dataset, consisting of 400 instructions in Chinese, spanning an extensive array of tasks such as creative writing, logical reasoning, semantic analysis, language comprehension, and mathematics. Each subject area was represented by 50 distinct samples. The evaluation process employed a pairwise comparison method, where human annotators were tasked with selecting the more suitable response from a pair, with the option to declare a tie. And we have set very lenient requirements for determining whether two responses can be labeled as a tie, so that the *win* could denote a clear advantage of one response over the other. To minimize variability, each sample is annotated by two evaluators. In this part, we mainly compare ChatGLM-32b-SFT and ChatGLM-32b-PPO, as PPO has shown an edge over DPO in automatic evaluation.

Results Table 4 shows the results of human evaluation and we report the win-rate of two models. The results reveal that the PPO model maintains a distinct advantage over the SFT model, with a win-rate of 30% compared to 15%. Regarding task-specific performance, the findings align with the observations from the automatic evaluations. Notably, the improvement in mathematical reasoning tasks are minimal and barely significant, whereas tasks related to writing exhibit considerable improvements. Moreover, an unexpected yet significant advancement was observed in programming tasks. However, based on our experience, it remains challenging for the reward model to accurately identify errors within code snippets. The programming instructions primarily focus on practical guidance, such as *how to build an Anaconda in linux*. This diverges from conventional code-evaluation benchmarks like HumanEval Chen et al. (2021), requiring the model to solve practical programming problems. Additionally, a high tie rate of 55% was recorded, aligning with our anticipations.

4.4 Evaluation on the Reward Model

We conduct experiments to evaluate the accuracy of the reward model in predicting human preferences. Specifically, we create a test set comprising pairwise comparisons of tasks from human evaluation, each accompanied by carefully-examined human annotations. At inference time, the reward model is capable of producing a scalar value for an individual response without the need for a paired sample. We utilize the reward model to predict the reward for each response and assess whether it assigns higher reward scores to the responses preferred by humans.

The results are presented in Table 5. We also include the accuracy of Llama2 reward model as a baseline for comparison. Despite ChatGLM and Llama2 being evaluated using distinct internal datasets, their results are still comparable as both models can reflect the alignment between human preferences and the predictions of the reward model. The data indicates that ChatGLM-32B surpasses ChatGLM-6B in terms of accuracy, as expected. However, the highest accuracy achieved by ChatGLM-32B is 65%, which is on par with that of Llama2-70B. The marginal accuracy gain with increased model scale suggests a limit to the benefits of scaling. Nonetheless, the fact that the reward

Table 5: Consistency between reward model and human preference on the internal test set. The results of Llama2 are from Touvron et al. (2023b). ChatGLM and Llama2 are not evaluated on the same test set. But the results could still be comparable in how accurately the reward model predicts human preferences. We report Accuracy.

	Training Acc	Test Acc
Llama2-7B	-	~0.61
Llama2-13B	-	~0.61
Llama2-70B	-	~0.64
ChatGLM-6B	0.64	0.59
ChatGLM-32B	0.68	0.65

Table 6: Response length of different models (Tokens) . Both DPO and PPO significantly increase the response length.

	ChatGLM-6B	ChatGLM-32B
SFT model	255.3	305.6
+ DPO	427.5	410.9 (+32.6%)
+ PPO	237.4	379.6 (+22.9%)

model can guide the training of RLHF algorithms with approximately 65% accuracy in mirroring human judgment is noteworthy and deserves further exploration.
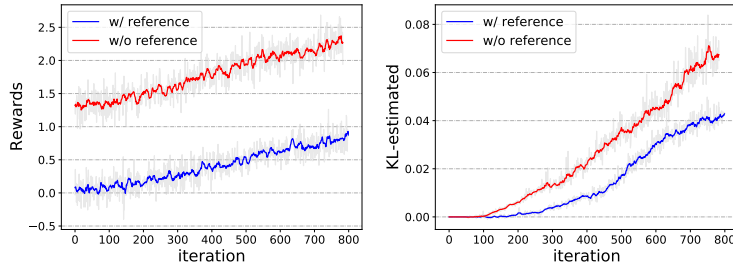

Figure 4: Ablation on the effects of reference reward. The two figures show the reward and estimated-KL during the training process of ChatGLM-32B-PPO.

4.5 Ablation Study

Response Length In automatic evaluation, GPT-4 serves as the automated evaluator for assessing response quality. Previous study Zheng et al. (2023a) suggests that GPT-4 is inclined to assign higher scores to lengthier responses, although the response may not imply better quality and instead denotes redundancy. To explore this aspect, we examine the lengths of responses generated by our models and report the results in Table 6. It is observed that all models produced responses that surpassed the SFT model in length. Notably, RLHF methods exhibit a remarkable increase in response length, i.e., 110 tokens in DPO and 75 tokens in PPO. Therefore, the improvement in automatic evaluation scores for creative writing tasks might be partly attributed to the increased response lengths. Moreover, the comparison between PPO and DPO reveals that despite the fact that PPO can generate shorter responses on average, it outperforms DPO in terms of performance. This indicates that PPO is capable of generating higher-quality responses with less redundancy.

Effects of Reference Rewards In our implementation of PPO, we employ a reference reward by deducting the reward of a baseline response. To evaluate the effectiveness of this approach, we compare PPO trained with (*w/*) and without (*w/o*) the reference reward, under the same hyper-parameter setting. Figure 4 illustrates the progression of reward and estimated KL-divergence throughout the training process. Remarkably, both experiments show similar patterns in reward improvement. However, the starting point of the reward value for the *w/ ref* scenario is near zero and this contributes to the stabilization of the training process, which is also observed in the training loss

curve. Moreover, the increase in estimated KL-divergence for the *w/o ref* scenario is much lower, thereby drifting less far from the original SFT model, as expected.

5 Conclusion

This paper introduces ChatGLM-RLHF, a reinforcement learning pipeline devised to enhance the alignment of ChatGLM with human preference. We provide an in-depth explanation of the three main components in ChatGLM-RLHF, data collection, reward model training, and policy model training. Additionally, we discuss the challenges encountered during the practices and the corresponding solutions, such as how to mitigate reward bias and variance, implement model parallelism, and avoid catastrophic forgetting. Experimental results indicate a significant improvement of ChatGLM-RLHF over ChatGLM-SFT, marking an average of 15% more wins in alignment with human preference, which underscores the efficacy of the proposed pipeline.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).

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

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073* (2022).

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 610–623.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism. *arXiv preprint arXiv:2401.02954* (2024).

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *NeurIPS* (2020).

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research* 24, 240 (2023), 1–113.

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* (2022).

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377* (2023).

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773* (2023).

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:247519241

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, et al. 2023. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459* (2023).

Alexander Havrilla, Maksym Zhuravinskyi, Duy Phung, Aman Tiwari, Jonathan Tow, Stella Biderman, Quentin Anthony, and Louis Castricato. 2023. trlX: A Framework for Large Scale Reinforcement Learning from Human Feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 8578–8595. https://doi.org/10.18653/v1/2023.emnlp-main.530

Zachary Kenton, Tom Everitt, Laura Weidinger, Iason Gabriel, Vladimir Mikulik, and Geoffrey Irving. 2021. Alignment of language agents. *arXiv preprint arXiv:2103.14659* (2021).

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, et al. 2023. Alignbench: Benchmarking chinese alignment of large language models. *arXiv preprint arXiv:2311.18743* (2023).

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *ICML* (2023).

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*.

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. *NeurIPS* (2022).

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* (2023).

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3505–3506.

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

Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Loose lips sink ships: Mitigating Length Bias in Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2310.05199* (2023).

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* (2023).

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. Preference ranking optimization for human alignment. *arXiv preprint arXiv:2306.17492* (2023).

Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Salmon: Self-alignment with principle-following reward models. *arXiv preprint arXiv:2310.05910* (2023).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

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

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. *ICLR* (2022).

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100* (2022).

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. 2023. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment. *arXiv preprint arXiv:2310.00212* (2023).

Zhewei Yao, Reza Yazdani Aminabadi, Olatunji Ruwase, Samyam Rajbhandari, Xiaoxia Wu, Ammar Ahmad Awan, Jeff Rasley, Minjia Zhang, Conglong Li, Connor Holmes, et al. 2023. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales. *arXiv preprint arXiv:2308.01320* (2023).

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2023. Glm-130b: An open bilingual pre-trained model. *ICLR* (2023).

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425* (2023).

Chenyu Zheng, Guoqiang Wu, Fan Bao, Yue Cao, Chongxuan Li, and Jun Zhu. 2023c. Revisiting discriminative vs. generative classifiers: Theory and implications. *arXiv preprint arXiv:2302.02334* (2023).

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023a. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685* (2023).

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. 2023b. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964* (2023).

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206* (2023).