# CHEEMS: A Practical Guidance for Building and Evaluating Chinese Reward Models from Scratch

Anonymous ACL submission

## Abstract

Reward models (RMs) are crucial for aligning large language models (LLMs) with human preferences. However, most RM research is centered on English and relies heavily on synthetic resources, which leads to limited and less reliable datasets and benchmarks for Chinese. To address this gap, we introduce CheemsBench, a fully human-annotated RM evaluation benchmark within Chinese contexts, and CheemsPreference, a large-scale and diverse preference dataset annotated through human-machine collaboration to support Chinese RM training. We systematically evaluate 20 RMs on CheemsBench and observe significant limitations in their ability to capture human preferences in Chinese scenarios. Additionally, based on CheemsPreference, we construct an RM that achieves state-of-the-art performance on CheemsBench, demonstrating the necessity of human supervision in RM training. Our findings reveal that scaled AI-generated data struggles to fully capture human preferences, emphasizing the importance of high-quality human supervision in RM development.

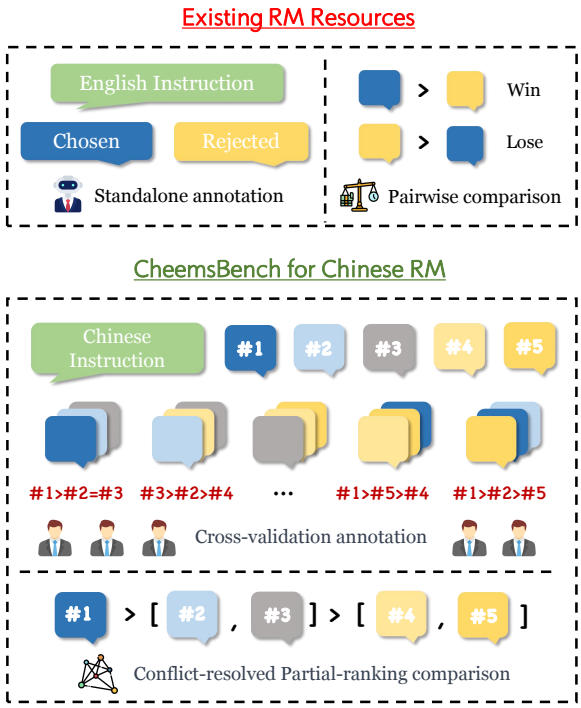



Figure 1: The differences in construction and usage between CheemsBench and the existing RM resources.

## 1 Introduction

With the rapid advancement of large language models (Yang et al., 2024; Dubey et al., 2024), post-training has emerged as a critical challenge for ensuring their safety, reliability, and alignment with human values (Hou et al., 2024; Lin et al., 2024). Reward models (Palan et al., 2019; Ouyang et al., 2022), as core components of LLM post-training, play a pivotal role in capturing human preferences and guiding models to adhere more closely to human needs (Bai et al., 2022). By providing reward signals, RMs can guide parameter optimization during training (Ibarz et al., 2018; Ouyang et al., 2022) or directly intervene outputs during decoding(Khanov et al., 2024; Li et al., 2024a).

Despite the crucial role of RMs in post-training, current research and resources are mainly focused on English. For instance, models such as Skywork-Reward (Liu et al., 2024a) and UltraRM (Cui et al., 2023) leverage high-quality English preference datasets (Zheng et al., 2023; Ji et al., 2024) and benchmarks (Lambert et al., 2024) to achieve superior performance. In contrast, the development of Chinese RMs faces significant challenges due to a lack of large-scale, high-quality preference datasets and comprehensive evaluation benchmarks. Moreover, existing RM resources mainly rely on synthetic data, which struggles to accurately reflect human preferences. Existing Chinese resources are often small in scale (Huozi-Team, 2024; Yucheng, 2023) and limited to specific domains (Zake, 2023; Xinlu Lai, 2024; Xu et al., 2023), thus insufficient for the needs of LLM alignment.

To address this critical gap, this paper con-

| Statistics | CheemsBench | | CheemsPreference | |
|---|---|---|---|---|
| | Open Prompt | Human Instruction | GPT | Human |
| # Prompts | 1,146 | 1,346 | 27,861 | 3,260 |
| # Responses | 5 | 5 | 5.29 | 5.07 |
| # Comparisons | 7,838 | 9,762 | 332,370 | 37,618 |
| Avg. Char. of Prompt | 186.58 | 197.04 | 175.56 | 164.08 |
| Avg. Char. of Chosen | 437.50 | 436.96 | 457.92 | 440.18 |
| Avg. Char. of Rejected | 454.01 | 446.43 | 394.18 | 432.84 |

Table 1: Statistics of CheemsBench and CheemsPreference: Number of prompts, average responses per prompt, comparisons (excluding ties), and average character lengths of prompts, chosen responses, and rejected responses.

structs a comprehensive and human-centric Chinese RM resource from scratch. It consists of two key datasets: (1) **CheemsBench**, a fully human-annotated and extensive Chinese RM evaluation benchmark that verifies whether RMs accurately capture and reflect human preferences; and (2) **CheemsPreference**, a large-scale, diverse Chinese preference dataset that provides supervised signals for training Chinese RMs, enabling them to effectively learn and model human preferences. [1]

As shown in Figure 1, unlike most RM resources that rely on machine-generated annotations (Zhou et al., 2024), CheemsBench and CheemsPreference are built on human supervision, thereby more accurately capturing realistic human values. Moreover, while traditional RM benchmarks (Lambert et al., 2024) typically rely on pairwise comparisons, recent studies (Wen et al., 2024) have highlighted their limitations in reflecting downstream performances. CheemsBench introduces a multi-response evaluation mechanism, which aligns closely with downstream tasks.

In CheemsBench, we combine open-source prompts and real-world human instructions with a comprehensive taxonomy to evaluate RM performance To better align with downstream tasks and reduce preference-induced noise (Zhang et al., 2024a), we sample five responses from various open- and closed-source LLMs for each prompt and conduct five rounds of human-driven triple-wise comparisons. To address potential annotation conflicts, we design a graph-based conflict-resolving algorithm that generates unique and consistent partial rankings. Using CheemsBench, we assess the progress of reward models and preference datasets in the Chinese context and identify considerable room for improvement in Chinese RMs.

For CheemsPreference, we collect 27k human instructions following a multi-tiered prompt taxonomy and sample more than 5 responses per prompt from various LLMs, ensuring both prompt and response diversity. To alleviate inconsistencies and biases in GPT annotations (Stureborg et al., 2024) while reducing human effort, we design a distant supervision algorithm to improve data quality. Specifically, human annotators first label a small golden preference dataset, which is then used to train an RM to filter a larger GPT-annotated dataset. The combined human- and GPT-annotated data form CheemsPreference, achieving state-of-the-art results on CheemsBench and performing well on the English RewardBench (Lambert et al., 2024).

Our contributions are summarized as follows:

- We propose CheemsBench, the first large-scale and comprehensive benchmark designed specifically for Chinese reward models.
- We construct CheemsPreference, the first large-scale, diverse, and high-quality Chinese preference dataset.
- We provide a comprehensive investigation into Chinese RM training and evaluation. The code and data will be publicly available at `https://github.com/XXX/XXX`.

## 2 Related Works

**Reinforcement Learning from Human Feedback.** Reinforcement Learning from Human Feedback has been widely adopted for LLM alignment (Ouyang et al., 2022; Bai et al., 2022). Previous research mostly focuses on specific tasks like summarization (Stiennon et al., 2022) and question answering (Nakano et al., 2022). Recent studies have expanded RLHF applications to broader domains (Hou et al., 2024; Lin et al., 2024; Yu et al., 2024), improving LLMs to be more helpful, honest, and harmless. RLHF enables models to align

[1] CHEEMS stands for *Chinese reward model benchmark and preference dataset*.

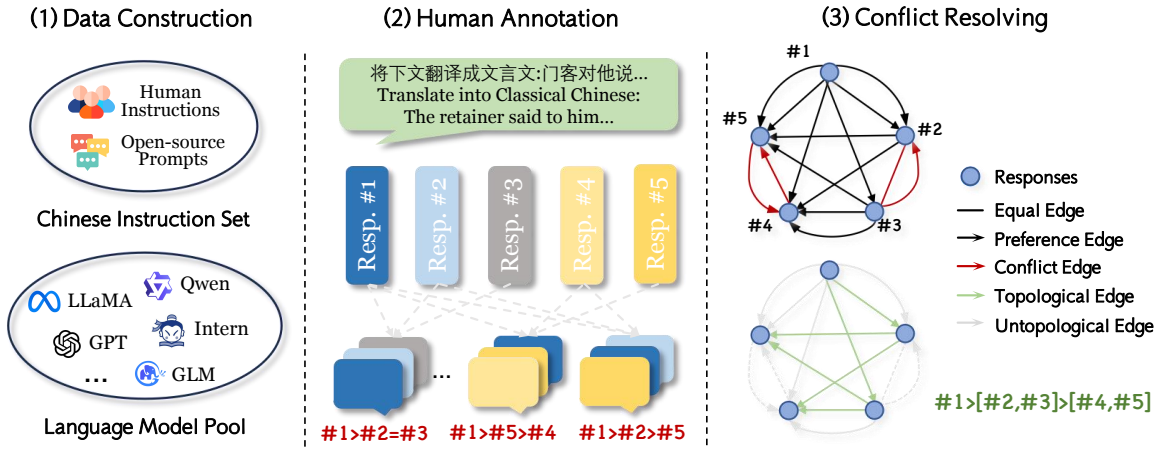



Figure 2: Chinese RM benchmark construction process. We utilize open-source prompts and human instructions and sample five responses from various models for each prompt. These responses then undergo five rounds of triple-wise manual comparisons. Unique partial rankings are generated by conflict resolving algorithm.

with human expectations more closely by integrating human preferences captured by reward models (Ng and Russell, 2000; Brown and Niekum, 2019; Palan et al., 2019). Thus, a reward model that accurately reflects human preferences is fundamental to the RLHF methodology.

**Reward Model Training and Evaluation.** Developing a RM that captures human preferences requires high-quality training datasets. Current works gather preference data through manual annotation (Bai et al., 2022; Zheng et al., 2023) or distilling advanced LLMs (Zhu et al., 2023; Cui et al., 2023). These works mostly focus on English, overlooking Chinese contexts. Existing Chinese preference datasets are generally small (Huozi-Team, 2024; Yucheng, 2023) or limited to specific tasks (Zake, 2023; Xinlu Lai, 2024; Xu et al., 2023). Beyond the training data, RM evaluation is also critical for post-training. The typical RM evaluation computes accuracy on a fixed test dataset (Lambert et al., 2024). Recent studies (Son et al., 2024; Kim et al., 2024; Zhou et al., 2024; Liu et al., 2024b; Frick et al., 2024; Gureja et al., 2024) have attempted to strengthen the correlation with downstream performance. However, these benchmarks focus on English, raising questions about their applicability to Chinese contexts.

## 3 Chinese RM Benchmark

In this section, we introduce **CheemsBench**, a benchmark designed to comprehensively evaluate Chinese RMs. Our benchmark is characterized by: *(1) High coverage*: We incorporate a wide range of prompts and sampling models, ensuring broad evaluation across diverse scenarios. *(2) High-quality annotation*: We derive a reliable preference ranking through multiple rounds of manual triple-wise comparisons and conflict resolving. Figure 2 illustrates the overall construction process.

### 3.1 Data Construction

**Prompt Collection.** We sample Chinese prompts from various open datasets, including Humaneval-XL (Peng et al., 2024), MathOctopus (Chen et al., 2024), GAOKAO-Bench (Zhang et al., 2024b), HalluQA (Cheng et al., 2023), Flames (Huang et al., 2023), CLiB (Lee, 2023), AlignBench (Liu et al., 2023), and COIG-CQIA (yuelin bai, 2023). We manually map their original categories into a unified system shown in Figure 8. We also include real-world human instructions for out-of-distribution evaluation. To ensure thorough converge across different scenarios, we build a comprehensive categorization system as illustrated in Figure 9. In total, we select 1,146 prompts from open-source datasets and 1,346 from human instructions.

**Responses Collection.** To ensure a wide range of response quality and distribution, we sample 5 responses per prompt from various models. (1) Open-source models: Qwen2-7B/72B-Instruct (Yang et al., 2024), Meta-Llama-3.1-8B/70B-Instruct (Dubey et al., 2024), Llama3.1-8B/72B-Chinese-Chat (Wang et al., 2024), Internlm2-chat-1.8b (Cai et al., 2024), and GLM-4-9b-chat (GLM et al., 2024); (2) Proprietary models: GPT-4 (OpenAI et al., 2024), GPT-3.5-turbo, GPT-4-turbo, and Claude-3-5-sonnet (Anthropic, 2024). We observe

that some open-source models demonstrate limited Chinese capabilities and tend to exhibit code-switching or even significant garbling[2]. In such cases, we rely on human annotators to filter these responses during the annotation process. Specifically, annotators are instructed to discard responses containing substantial sections of meaningless content, while retaining those with minor code-switching that do not compromise semantic meaning. This procedure allows us to account for LLMs' code-switching behavior during RM evaluation.

### 3.2 Benchmark Labeling

**Human Annotation.** To accurately capture human preferences, CheemsBench relies entirely on human judgment for its annotation process. Given a prompt and its corresponding 5 responses, we pre-design five annotation tasks, each comprising a triple-wise comparison of three adjacent responses. These tasks are distributed to different annotators who perform preference comparisons independently. All annotation results are then used to construct a ranked list of responses.

**Conflict Resolving.** However, conflicts may arise due to the human preferences ambiguity and potential annotation errors. To derive reliable results, we develop a dedicated conflict resolving algorithm, as shown in Algorithm 1. Specifically, we first transform the annotation results into a directed preference graph, where responses and preferences represent nodes and edges respectively. We then employ depth-first search to identify cycles in the graph, which indicate conflicts. These cycles are merged into larger nodes, and this process is repeated until no cycles remain in the graph. Finally, we perform topological sorting to obtain a partial ranking. [3]

### 3.3 Evaluation Metrics

Given that we have multiple responses per prompt, there are many potential metrics for evaluation (Wen et al., 2024). We first convert a partial ranking into multiple pair-wise comparisons and evaluate the accuracy following typical setting (Lambert et al., 2024):

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(r_w^i > r_l^i) \quad (1)$$

where $N$ is the total number of pair-wise comparisons after transformation, and the indicator function $\mathbb{I}$ checks if the reward score for the preferred response $r_w^i$ is greater than that of its counterpart $r_l^i$. Additionally, the exact match rate can be employed, which measures the proportion of prompts where all pair-wise comparisons are correctly sorted:

$$\text{Exact Match} = \frac{1}{M}\sum_{j=1}^{M}\mathbb{I}\left(\bigwedge_k (r_w^{j,k} > r_l^{j,k})\right) \quad (2)$$

where $M$ is the number of prompts, and the indicator function verifies if all comparisons are ordered correctly. We obtain the final result by averaging the metrics from subsets of different categories.

## 4 Chinese Preference Dataset

In this section, we present the construction of **CheemsPreference**, as depicted in Figure 3. Our dataset is characterized by: *(1) Scale and diversity*: We amass 27k real human instructions, featuring a comprehensive multi-tier categorization system, and sample multiple responses from a variety of models for each prompt. *(2) High-quality annotation*: We employ a distant supervision algorithm, which integrates both human annotations and GPT-4o to establish reliable partial preference ranks.

### 4.1 Data Construction

**Prompt Collection.** Diverse and high-quality instruction data are crucial for ensuring the robustness of RMs. To this end, we collect 27,861 real-world human instructions. To ensure extensive coverage of downstream scenarios, we develop a comprehensive multi-tier categorization system, which encompasses eight main categories with dozens of refined subcategories, as illustrated in Figure 10.

**Response Collection.** We sample responses from a broad range of models: (1) Open-source models: Qwen2-7B/72B-Instruct (Yang et al., 2024), Qwen2.5-7B/14B/32B/72B-Instruct (Team, 2024), Meta-Llama-3.1-8B/70B-Instruct (Dubey et al., 2024), Llama3.1-8B/72B-Chinese-Chat (Wang et al., 2024), Internlm2-chat-1.8b (Cai et al., 2024), and GLM-4-9b-chat (GLM et al., 2024). (2) Proprietary models: GPT-4 (OpenAI et al., 2024), GPT-3.5-turbo, GPT-4-turbo, GPT-4o, and Claude-3-5-sonnet (Anthropic, 2024). To guarantee the quality of responses, we implement rule-based methods to detect responses that are abnormally lengthy or contain excessive non-Chinese symbols. Finally, each prompt has more than 5 responses on average.

[2]The LLaMA series shows a higher tendency for code-switching and nonsensical output, possibly due to its tokenizer vocabulary and insufficient training on Chinese corpora.

[3]Details about the algorithms and annotators are provided in Appendix C and Appendix D, respectively.

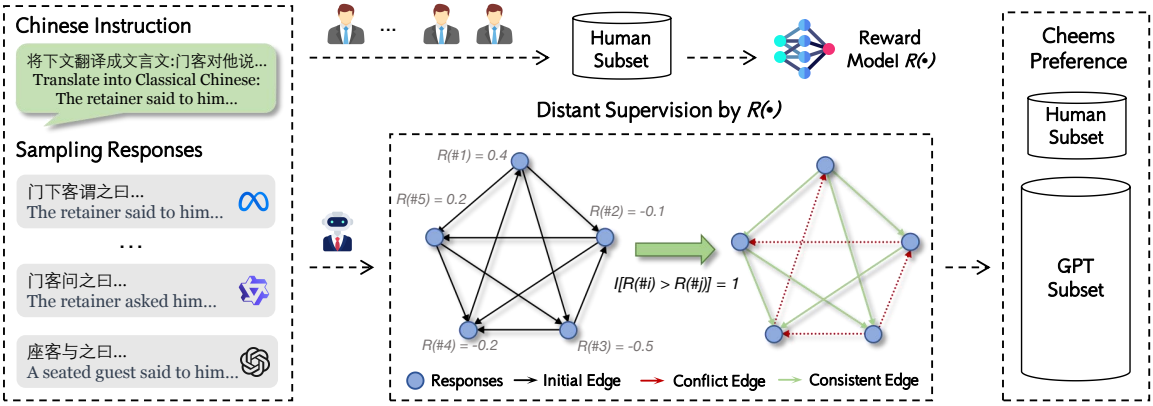



Figure 3: Chinese preference dataset construction process. Each prompt's different responses and their annotation results form a directed graph. Circles in this preference graph indicate conflicts. We utilize the reward model trained on the human-annotated dataset to filter GPT annotations, thereby producing a directed acyclic graph.

### 4.2 Distant Supervision

The quality of preference data (Gao et al., 2024) is essential for the training of RM. While human annotation ensures high quality, it is expensive and challenging to obtain in large quantities. Conversely, GPT-based annotation is scalable but often inconsistent and biased (Stureborg et al., 2024). To construct large-scale, high-quality Chinese preference data, we implement a distant supervision strategy for annotation. We initially engage human annotators to label a small subset of data, following the protocol detailed in Section 3.2. Subsequently, GPT-4o is employed to annotate a larger dataset. For a set of $N$ responses, GPT-4o performs $C_N^2$ pair-wise comparisons between each response pairs[4]. To mitigate positional bias (Li et al., 2024b), the order of responses in each comparison is randomized. Although these GPT-4o annotations can exhibit inconsistencies, i.e., cycles in the preference graph, we employ an RM trained on human-annotated data to filter these annotations and establish a consistent partial order. Additionally, we propose a length-debias post-hoc filtering strategy to alleviate length bias (Dubois et al., 2024). This involves dividing the dataset into two groups, where the chosen response is longer or shorter than the rejected one, and downsampling the larger group to balance the dataset.

## 5 Chinese Reward Model

In this section, we introduce our reward model training methodology. In contrast to typical preference datasets constructed by pair-wise comparisons (Cui et al., 2023; Ji et al., 2024), CheemsPreference has two distinct characteristics: *(1) each prompt is associated with multiple responses*, and *(2) these responses form only a partial preference chain*. Thus, we employ following loss according to Bradley-Terry Model (Bradley and Terry, 1952):

$$\mathcal{L}' = - \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{X} \\ y_w,y_l\sim\mathcal{Y}_x}} \left[\log\left(\sigma\left(r\left(x,y_w\right) - r\left(x,y_l\right)\right)\right)\right] \tag{3}$$

where $\mathcal{X}$ stands for the distribution of the prompt $x$ and $\mathcal{Y}_x$ denotes the distribution of responses $y$ given the prompt $x$. We employ a greedy sample-based batch logic for calculating this loss. Specifically, during each forward pass, we determine if all responses for a given prompt can be included in one batch. If feasible, they are added to the batch; otherwise, any excess responses are allocated to subsequent batches. This method might bypass some pair comparisons, but it ensures that no response is duplicated across batches, thereby mitigating overfitting risks (Ouyang et al., 2022). More importantly, this sample-based batch organization enhances computational efficiency by reducing redundant forward passes. To further stabilize training, we integrate an additional regularization term (Hou et al., 2024), imposing a Gaussian prior on the distribution of reward scores:

$$\mathcal{L} = \mathcal{L}' + \mathop{\mathbb{E}}_{x\sim\mathcal{X},y\sim\mathcal{Y}_x} \left[r^2\left(x,y\right)\right] \tag{4}$$

## 6 Experiments

We first assess the performance of open-source RMs and datasets on CheemsBench (see Section 6.1). Next, we examine our benchmark's correlation with downstream tasks (Section 6.2). For

[4]Annotation prompts can be found in Appendix B.

| Model Name | RewardBench | Open Prompt | | Human Instruction | | Overall |
|---|---|---|---|---|---|---|
| | | Acc. | Exact. | Acc. | Exact. | |
| ***Open-source Reward Models*** | | | | | | |
| Skywork-Reward-Gemma-2-27B | 0.938 | 0.754 | 0.329 | 0.748 | 0.311 | 0.535 |
| Skywork-Reward-Gemma-2-27B-v0.2 | 0.943 | 0.751 | 0.321 | 0.735 | 0.294 | 0.525 |
| Llama-3.1-Nemotron-70B-Reward-HF | 0.941 | 0.750 | 0.317 | 0.722 | 0.271 | 0.515 |
| Llama-3-OffsetBias-RM-8B | 0.894 | 0.734 | 0.310 | 0.689 | 0.239 | 0.493 |
| RM-Mistral-7B | 0.804 | 0.721 | 0.285 | 0.700 | 0.259 | 0.491 |
| URM-LLaMa-3-8B | 0.899 | 0.727 | 0.310 | 0.688 | 0.230 | 0.489 |
| ArmoRM-Llama3-8B-v0.1 | 0.904 | 0.715 | 0.308 | 0.677 | 0.246 | 0.487 |
| Skywork-Reward-Llama-3.1-8B-v0.2 | 0.931 | 0.721 | 0.283 | 0.701 | 0.237 | 0.486 |
| URM-LLaMa-3.1-8B | 0.929 | 0.722 | 0.292 | 0.696 | 0.230 | 0.485 |
| GRM-Llama3-8B-rewardmodel-ft | 0.915 | 0.728 | 0.281 | 0.688 | 0.229 | 0.482 |
| ***Generative Models as Reward Models*** | | | | | | |
| Skywork-Critic-Llama-3.1-70B | 0.933 | 0.755 | 0.320 | 0.731 | 0.258 | 0.516 |
| CompassJudger-1-14B-Instruct | 0.841 | 0.745 | 0.327 | 0.692 | 0.239 | 0.501 |
| Qwen2.5-72B-Instruct | - | 0.734 | 0.306 | 0.678 | 0.229 | 0.487 |
| Skywork-Critic-Llama-3.1-8B | 0.890 | 0.726 | 0.288 | 0.696 | 0.229 | 0.485 |
| GPT-4o | 0.846 | 0.640 | 0.163 | 0.727 | 0.300 | 0.457 |
| Doubao-pro-128k | - | 0.720 | 0.280 | 0.662 | 0.164 | 0.456 |
| Qwen2.5-7B-Instruct | - | 0.713 | 0.262 | 0.637 | 0.163 | 0.444 |
| Llama-3-OffsetBias-8B | 0.840 | 0.690 | 0.243 | 0.658 | 0.180 | 0.443 |
| Llama-3.1-70B-Instruct | 0.840 | 0.685 | 0.244 | 0.610 | 0.153 | 0.423 |
| CompassJudger-1-1.5B-Instruct | 0.734 | 0.660 | 0.210 | 0.594 | 0.132 | 0.399 |

Table 2: Performance of discriminative and generative RMs on CheemsBench. The **Overall** metric is the average of accuracy (**Acc.**) and exact match (**Exact.**) across the Open Prompt and Human Instruction subsets.

CheemsPreference, we conduct an ablation study to demonstrate its effectiveness (Section 6.3) and test the scaling trend (Section 6.4).

### 6.1 Benchmark Results

**Reward Models Evaluation** We thoroughly assess the performance of current RMs in the Chinese context, including discriminative reward models and generative models as reward models (Zheng et al., 2023). Table 2 demonstrates the results of top-ranked RMs on CheemsBench. We find that (1) **The accuracy of the leading models significantly drops when applied to CheemsBench.** This performance gap indicates considerable room for improvement of RMs in Chinese settings. (2) **These RMs perform better on open-source prompts than on human instructions**. This is expected, as our human instructions are collected from the real world and thus can be more out-of-distribution than open-source prompts. (3) **For prompts with relatively deterministic answers, RM can assess the quality of the responses more accurately.** Figure 4 details the performance of these RMs on different subcategories. On the open-source prompt subset, RMs show competence in "Reasoning" but struggle in other categories. On the human instruction subset, models excel in "Reasoning" and "Complex Instructions" but perform poorly in tasks involving "Understanding". These observations emphasize the need for targeted enhancements in these tasks.

**Preference Datasets Evaluation** We evaluate various Chinese and English preference datasets on CheemsBench by training RMs[5] based on Qwen2.5-72B-Instruct (Team, 2024). The experimental results are presented in Table 3. Notably, among the Chinese datasets, "Huozi" (Huozi-Team, 2024) performs best. Meanwhile, the "Ultrafeedback" (Cui et al., 2023) leads among English datasets. Comparisons of the top-performing English and Chinese preference datasets on CheemsBench reveal **a critical gap between English and Chinese preference datasets**, which highlights a need for better Chinese preference dataset.

### 6.2 Downstream Correlation

In this section, we explore the correlation of CheemsBench with various downstream tasks by employing a Best-of-32 sampling strategy for optimization. We evaluate three downstream tasks: Human Win-rate, MT-bench-zh (Huozi-Team, 2024), and MT-bench (Zheng et al., 2023). For the Hu-

[5]Details about hyperparameter settings for different experiments are provided in Appendix F.

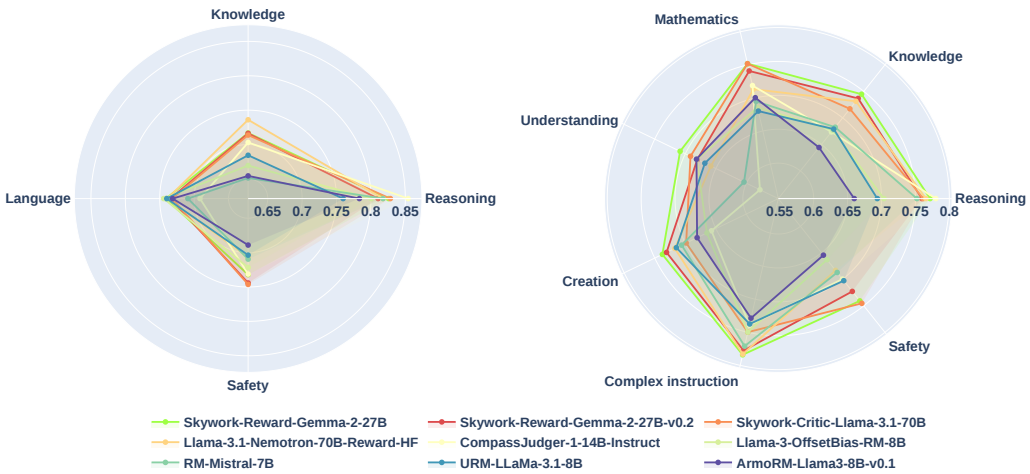



Figure 4: Accuracy of top-ranked reward models on CheemsBench across subsets of different categories. The left and right sub-figures respectively show the results on open-source prompts and human instructions.

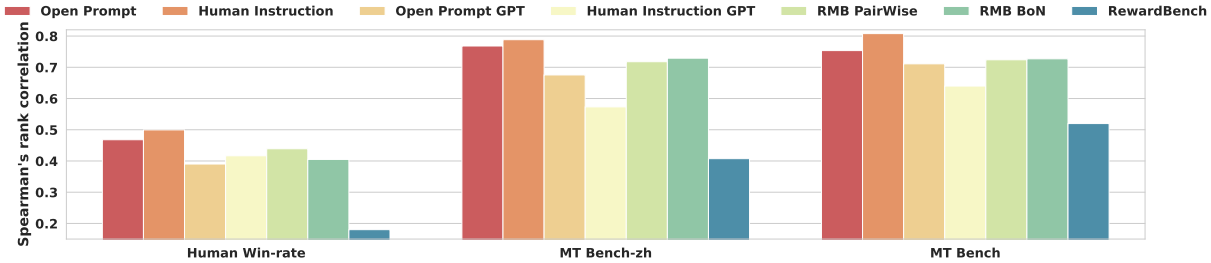



Figure 5: Correlations between different RM benchmarks an performance on three downstream tasks.

| Dataset | Open Prompt | | Human Instruction | |
|---|---|---|---|---|
| | Acc. | Exact. | Acc. | Exact. |
| ***Chinese Preference Datasets*** | | | | |
| HH-RLHF-cn | 0.704 | 0.306 | 0.646 | 0.212 |
| Huozi | 0.728 | 0.302 | 0.682 | 0.237 |
| Kyara | 0.705 | 0.258 | 0.664 | 0.198 |
| Zhihu | 0.463 | 0.105 | 0.487 | 0.080 |
| ***English Preference Datasets*** | | | | |
| ChatbotArena | 0.745 | 0.342 | 0.718 | 0.288 |
| HH-RLHF | 0.753 | 0.351 | 0.740 | 0.299 |
| MathPreference | 0.566 | 0.179 | 0.502 | 0.103 |
| Nectar | 0.716 | 0.288 | 0.664 | 0.222 |
| PKU-SafeRLHF | 0.737 | 0.311 | 0.678 | 0.240 |
| Skywork | 0.757 | 0.343 | 0.749 | 0.271 |
| MathStackExchange | 0.749 | 0.340 | 0.719 | 0.256 |
| UltraFeedback | 0.768 | 0.356 | 0.748 | 0.303 |

Table 3: Performance results of various datasets. Each dataset's performance is evaluated under Open Prompt and Human Instruction subsets, with results presented in terms of accuracy (Acc.) and exact match (Exact.).

man Win-rate task, we use 87 unique Chinese instructions that are not included in our benchmark. For each instruction, we obtain a fixed baseline response from Qwen2-72B-Instruct. Then, we sample 32 additional responses from the same model and have human annotators score each one, assigning 1 if a response exceeds the baseline and -1 if it doesn't. This allows us to determine win rates for each RM using the Best-of-32 strategy. For MT-bench-zh and MT-bench, responses are sampled from Qwen2-7B-Instruct, with RMs performing Best-of-32 sampling on two-turn prompts, and GPT-4o is employed as the judge. We select 26 distinct open reward models, differing in training data and structures, for correlation assessment. Our baselines include *RewardBench* (Lambert et al., 2024), *RMB* (Zhou et al., 2024), and alternatives of our benchmarks annotated by GPT-4o, named as *Open Prompt GPT* and *Human Instruction GPT*. The results in Figure 5 illustrate that: (1) **Our benchmark exhibits significantly stronger correlations with downstream tasks** compared to other baselines, whether in Chinese or English tasks. (2) **The benchmarks annotated by GPT demonstrate suboptimal correlation**, underscoring the necessity of human judgment, which can achieve better generalization on downstream tasks.

### 6.3 Dataset Construction Ablation

We conduct an ablation study to assess the effectiveness of the dataset construction strategies outlined

| Model | RewardBench | Open Prompt | | Human Instruction | | Overall |
|---|---|---|---|---|---|---|
| | | Acc. | Exact. | Acc. | Exact. | |
| ***State-of-the-art Baselines*** | | | | | | |
| RewardBench@1 | **0.943** | 0.751 | 0.321 | 0.735 | 0.294 | 0.525 |
| RewardBench@2 | 0.941 | 0.750 | 0.317 | 0.722 | 0.271 | 0.515 |
| ***Models trained using CheemsPreference*** | | | | | | |
| Human subset | 0.897 | 0.852 | 0.502 | 0.823 | 0.412 | 0.647 |
| GPT subset | 0.822 | 0.778 | 0.373 | 0.743 | 0.303 | 0.549 |
| w/ Length debiasing | 0.865 | 0.790 | 0.402 | 0.768 | 0.322 | 0.571 |
| w/ Distant supervision | 0.909 | 0.837 | 0.464 | 0.821 | 0.404 | 0.632 |
| w/ All strategies | 0.917 | 0.837 | 0.458 | 0.826 | 0.416 | 0.634 |
| CheemsPreference | 0.919 | **0.857** | **0.508** | **0.832** | **0.431** | **0.657** |

Table 4: The performance of RMs trained on our datasets, along with ablation studies on different processing strategies. CheemsPreference represents a combination of the fully processed GPT subset with the human subset.

in Section 4.2. We train RMs based on Qwen2.5-72b-instruct (Team, 2024) to perform experiments and report performances in Table 4. The results reveal several key insights: (1) **Neither *Human* nor *GPT* subsets alone are sufficient.** The *GPT* subset underperforms on our benchmark, indicating the inability of GPT-4o to fully capture human preferences. Conversely, the *Human* subset performs poorly on RewardBench, likely due to its smaller scale, which limits out-of-distribution performance. (2) **Length-debias strategy enhances performance.** We investigate the biases of GPT and human annotators in Appendix E, highlighting the necessity of a length-debias strategy. (3) **Distant supervision strategy significantly improves performance**, highlighting the importance of incorporating human supervision. (4) **The integration of all strategies performs the best**, underscoring the effectiveness of our approach.

### 6.4 Scaling Trend

We validate scaling trends on CheemsPreference. Figure 6 shows that RM performance improves with increased data volume on Open Prompt and Human Instruction subsets, indicating that **larger training dataset leads to superior performance**. This phenomenon also highlights the potential of our distant supervision approach. We then assess model scaling trending by training RM on different sizes of Qwen-2.5 series models (Team, 2024). Figure 7 illustrates that increasing the model size from 0.5B to 72B significantly enhances performance, demonstrating that **larger models capture complex preference patterns more effectively**. Moreover, there is no significant difference when starting training from pretrained or instruct models.

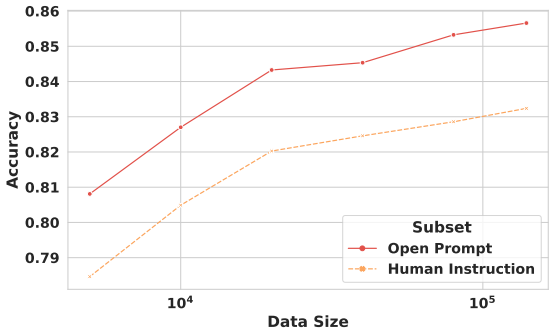



Figure 6: Impact of data size scaling measured by the number of pairs on accuracy.

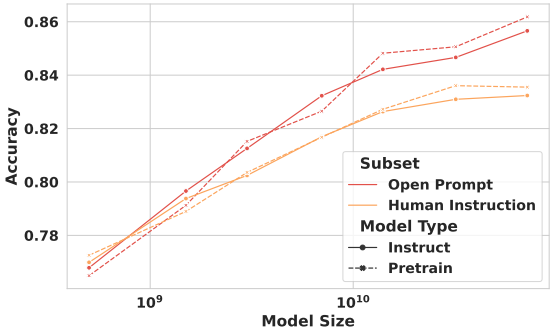



Figure 7: Impact of model size scaling on RM accuracy.

## 7 Conclusion

In this paper, we address the challenges of developing Chinese RMs by introducing CheemsBench, a comprehensive RM benchmark, and CheemsPreference, a high-quality Chinese preference dataset. Using these resources, we evaluate the progress of RMs in the Chinese context and validate the effectiveness of our dataset construction strategies. Our work narrows the gap between English and Chinese RMs and sets the foundation for future research.

## Limitations

This work addresses the resource insufficiency in Chinese reward models. However, by focusing primarily on the Chinese language, the datasets may not fully capture all regional variations, potentially introducing language and cultural biases. Additionally, while the importance of human annotations is evident, the subjective nature of human judgment and the particular group of annotators involved can lead to biased preferences. Moreover, our findings, while tailored to the Chinese context, require further validation to ensure applicability beyond Chinese and English languages.

## Ethical Considerations

Several ethical considerations are central to this work. Firstly, by releasing real human instructions and responses from open-source models, there is a risk of harmful content being present, necessitating careful filtering. Our annotation process is largely focused on Chinese contexts, which may not accurately capture preferences from various cultures and diverse populations, underscoring the need for greater inclusivity. Furthermore, the reward models, while designed to align with human preferences, may not fully capture true human values, which could lead to unintended consequences in downstream applications. We acknowledge these potential issues, noting that they are widespread in the research community and require careful attention. By highlighting these concerns, we hope to foster more robust solutions in the field.

## References


Anthropic. 2024. Claude 3.5 sonnet. https://www.anthropic.com/claude/sonnet.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

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

Daniel S. Brown and Scott Niekum. 2019. Deep bayesian reward learning from preferences. *Preprint*, arXiv:1912.04472.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. 2024. Internlm2 technical report. *Preprint*, arXiv:2403.17297.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. 2024. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. *Preprint*, arXiv:2310.20246.

Qinyuan Cheng, Tianxiang Sun, Wenwei Zhang, Siyin Wang, Xiangyang Liu, Mozhi Zhang, Junliang He, Mianqiu Huang, Zhangyue Yin, Kai Chen, and Xipeng Qiu. 2023. Evaluating hallucinations in chinese large language models. *CoRR*, abs/2310.03368.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, and other. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *Preprint*, arXiv:2404.04475.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios N. Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024.

How to evaluate reward models for rlhf. *Preprint*, arXiv:2410.14872.

Yang Gao, Dana Alon, and Donald Metzler. 2024. Impact of preference noise on the alignment performance of generative language models. *Preprint*, arXiv:2404.09824.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Srishti Gureja, Lester James V. Miranda, Shayekh Bin Islam, Rishabh Maheshwary, Drishti Sharma, Gusti Winata, Nathan Lambert, Sebastian Ruder, Sara Hooker, and Marzieh Fadaee. 2024. M-rewardbench: Evaluating reward models in multilingual settings. *Preprint*, arXiv:2410.15522.

Zhenyu Hou, Yilin Niu, Zhengxiao Du, Xiaohan Zhang, Xiao Liu, Aohan Zeng, Qinkai Zheng, Minlie Huang, Hongning Wang, Jie Tang, and Yuxiao Dong. 2024. Chatglm-rlhf: Practices of aligning large language models with human feedback. *Preprint*, arXiv:2404.00934.

Kexin Huang, Xiangyang Liu, Qianyu Guo, Tianxiang Sun, Jiawei Sun, Yaru Wang, Zeyang Zhou, Yixu Wang, Yan Teng, Xipeng Qiu, Yingchun Wang, and Dahua Lin. 2023. Flames: Benchmarking value alignment of chinese large language models. *Preprint*, arXiv:2311.06899.

Huozi-Team. 2024. Huozi: Leveraging large language models for enhanced open-domain chatting. https://github.com/HIT-SCIR/huozi.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. 2018. Reward learning from human preferences and demonstrations in atari. *Preprint*, arXiv:1811.06521.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. 2024. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. *arXiv preprint arXiv:2406.15513*.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. 2024. Args: Alignment as reward-guided search. *Preprint*, arXiv:2402.01694.

Sunghwan Kim, Dongjin Kang, Taeyoon Kwon, Hyungjoo Chae, Jungsoo Won, Dongha Lee, and Jinyoung Yeo. 2024. Evaluating robustness of reward models for mathematical reasoning. *Preprint*, arXiv:2410.01729.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. *Preprint*, arXiv:2403.13787.

Jein Lee. 2023. chinese-llm-benchmark. https://github.com/jeinlee1991/chinese-llm-benchmark.

Bolian Li, Yifan Wang, Ananth Grama, and Ruqi Zhang. 2024a. Cascade reward sampling for efficient decoding-time alignment. *Preprint*, arXiv:2406.16306.

Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2024b. Split and merge: Aligning position biases in llm-based evaluators. *Preprint*, arXiv:2310.01432.

Mingan Lin, Fan Yang, Yanjun Shen, Haoze Sun, Tianpeng Li, Tao Zhang, Chenzheng Zhu, Tao Zhang, Miao Zheng, Xu Li, Yijie Zhou, Mingyang Chen, Yanzhao Qin, Youquan Li, Hao Liang, Fei Li, Yadong Li, Mang Wang, Guosheng Dong, Kun Fang, Jianhua Xu, Bin Cui, Wentao Zhang, Zenan Zhou, and Weipeng Chen. 2024. Baichuan alignment technical report. *Preprint*, arXiv:2410.14940.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024a. Skywork-reward: Bag of tricks for reward modeling in llms. *Preprint*, arXiv:2410.18451.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, Xiaohan Zhang, Lichao Sun, Hongning Wang, Jing Zhang, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. Alignbench: Benchmarking chinese alignment of large language models. *Preprint*, arXiv:2311.18743.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2024b. Rm-bench: Benchmarking reward models of language models with subtlety and style. *Preprint*, arXiv:2410.16184.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for inverse reinforcement learning. In *Proceedings*

*of the Seventeenth International Conference on Machine Learning*, ICML '00, page 663–670, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Malayandi Palan, Nicholas C. Landolfi, Gleb Shevchuk, and Dorsa Sadigh. 2019. Learning reward functions by integrating human demonstrations and preferences. *Preprint*, arXiv:1906.08928.

Qiwei Peng, Yekun Chai, and Xuhong Li. 2024. Humaneval-xl: A multilingual code generation benchmark for cross-lingual natural language generalization. *Preprint*, arXiv:2402.16694.

Guijin Son, Hyunwoo Ko, Hoyoung Lee, Yewon Kim, and Seunghyeok Hong. 2024. Llm-as-a-judge & reward model: What they can and cannot do. *Preprint*, arXiv:2409.11239.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback. *Preprint*, arXiv:2009.01325.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *Preprint*, arXiv:2405.01724.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Shenzhi Wang, Yaowei Zheng, Guoyin Wang, Shiji Song, and Gao Huang. 2024. Llama3.1-8b-chinese-chat.

Xueru Wen, Jie Lou, Yaojie Lu, Hongyu Lin, Xing Yu, Xinyu Lu, Ben He, Xianpei Han, Debing Zhang, and Le Sun. 2024. Rethinking reward model evaluation: Are we barking up the wrong tree? *Preprint*, arXiv:2410.05584.

shareAI Xinlu Lai. 2024. The dpo dataset for chinese and english with emoji. `https://huggingface.co/datasets/shareAI/DPO-zh-en-emoji`.

Guohai Xu, Jiayi Liu, Ming Yan, Haotian Xu, Jinghui Si, Zhuoran Zhou, Peng Yi, Xing Gao, Jitao Sang, Rong Zhang, Ji Zhang, Chao Peng, Fei Huang, and Jingren Zhou. 2023. Cvalues: Measuring the values of chinese large language models from safety to responsibility. *Preprint*, arXiv:2307.09705.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Huimu Yu, Xing Wu, Weidong Yin, Debing Zhang, and Songlin Hu. 2024. Codepmp: Scalable preference model pretraining for large language model reasoning. *Preprint*, arXiv:2410.02229.

Li Yucheng. 2023. 3,000 chinese zhihu q&a preference dataset. https://huggingface.co/datasets/liyucheng/zhihu_rlhf_3k.

yuelin bai. 2023. Coig-cqia: Quality is all you need for chinese instruction fine-tuning. https://github.com/paralym/COIG-CQIA.

Zake. 2023. Kyara: Knowledge yielding adaptive retrieval augmentation. https://github.com/zake7749/kyara/.

Michael JQ Zhang, Zhilin Wang, Jena D. Hwang, Yi Dong, Olivier Delalleau, Yejin Choi, Eunsol Choi, Xiang Ren, and Valentina Pyatkin. 2024a. Diverging preferences: When do annotators disagree and do models know? *Preprint*, arXiv:2410.14632.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2024b. Evaluating the performance of large language models on gaokao benchmark. *Preprint*, arXiv:2305.12474.

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. *Preprint*, arXiv:2306.05685.

Enyu Zhou, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, Yurong Mou, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Rmb: Comprehensively benchmarking reward models in llm alignment. *Preprint*, arXiv:2410.09893.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. 2023. Starling-7b: Improving llm helpfulness & harmlessness with rlaif.

## A Category System

The prompt category taxonomy for CheemsBench is illustrated in Figure 8 to 9, while the promot category taxonomy for CheemsPreference is illustrated in Figure 10.

## B Annotation Prompts

In this work, we leverage GPT-4o for constructing our preference dataset. We utilize the structured judge prompt presented in Figure 11 to assess response quality, emphasizing an objective and unbiased comparison between different model outputs. Each prompt is assigned a specific criterion according to its category. These criteria ensure that the evaluations are consistent and comprehensive across different contexts. Figure 13 provides a detailed overview of the criteria in Chinese, covering linguistic and logical aspects. It also accounts for the safety and complexity of instructions. [6]

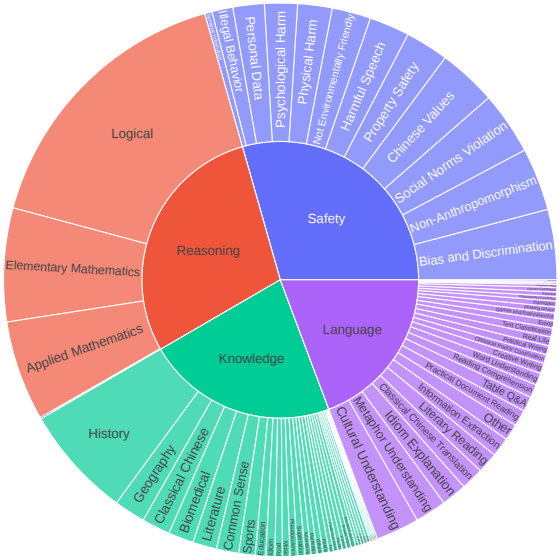



Figure 8: Category system for open-source prompts, which are selected from various datasets and manually integrated into this unified framework.

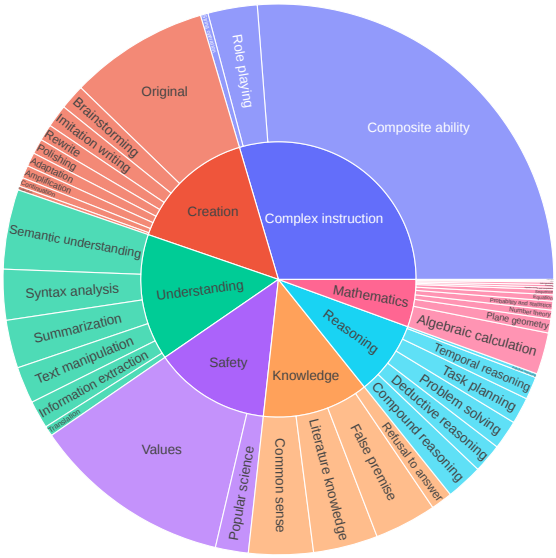



Figure 9: Category system for human instructions. Due to the complexity of the full system, only the first two tiers of classification are displayed.

## C Conflict Resolving

In this section, we introduce an algorithm designed to address potential annotation conflicts that arise from human evaluations. The Conflict Resolving

[6] The English versions of the judge prompt template and criteria are displayed in Figure 12 and 14.

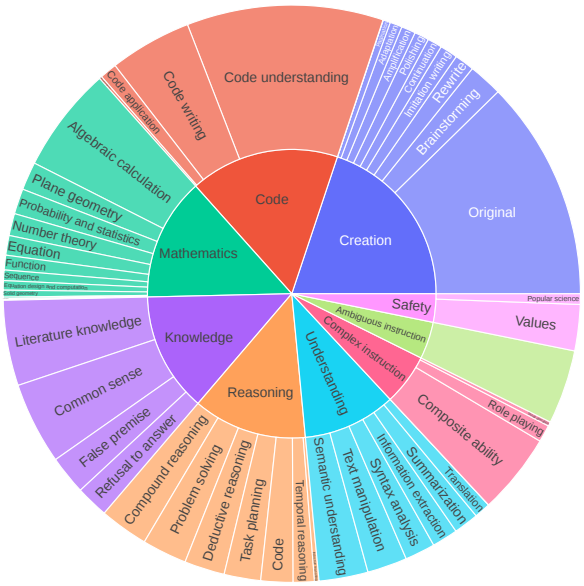



Figure 10: Category system for prompts in the Chinese Preference Dataset. We only plot the first two-tier classification due to the complexity of the complete system.

Algorithm, as outlined in Algorithm 1, operates by systematically integrating conflicting responses into larger nodes, based on the understanding that these responses exhibit comparable quality. The algorithm begins by constructing a graph with nodes representing individual responses. Directed edges are established based on preference relationships between responses. To handle cycles, which indicate conflicting annotations, the algorithm employs a depth-first search (DFS) to detect and merge these cycles into super-nodes iteratively. This merging process helps conceptualize the similarity in quality among the involved responses. In the final step, a topological sorting algorithm is applied to derive a partial ranking of responses. We report the conflict rate between human annotations and GPT annotations on the Open Prompts and Human Instruction subsets in Table 5. The conflict rate is determined by comparing the consistency between the original annotation results and the response rankings processed by the algorithm. We find that, overall, GPT is more inconsistent than human annotators. Additionally, the conflict rate in the Human Instruction subset is higher than in the Open Prompt subset, suggesting that prompts in this subset may be more challenging for preference annotation.

## D Human Annotation Details

We employ a team of 29 annotators, each holding a bachelor's degree. On average, an annotator completes approximately 40 triple-wise comparisons per day. Annotation tasks are assigned using a system that guarantees that each annotator receives unique data. During the process, annotators have the flexibility to re-assign tasks they find challenging to other team members, thereby improving the efficiency of data annotation. To ensure high-quality results, we have additional quality assurance personnel and reviewers who assess the consistency of the data. Data are only finalized and delivered if the consistency among annotators, quality assurance personnel, and reviewers exceeds 90%. These procedures are in place to uphold the integrity and quality of our data.

Table 5: Conflict ratio of human annotations and GPT-4o annotations.

| Dataset | Conflict Ratio |
|---|---|
| Open Prompt Human | 0.1999 |
| Human Instruction Human | 0.2161 |
| Open Prompt GPT | 0.2593 |
| Human Instruction GPT | 0.3170 |

## E Annotation Bias

We also explore the preferences of both human and GPT annotators in terms of response length and position, as shown in Figure 15. It can be observed that GPT-4o generally prefers responses that are placed later, whereas human annotators do not exhibit a significant preference for position. Additionally, when the response length difference is moderate, both human and GPT annotators tend to favor longer responses. However, as the length difference becomes too large, humans tend to prefer shorter ones. Overall, the specific preferences of the annotators are not very pronounced.

| Hyperparameter | Value |
|---|---|
| Max Sequence Length | 2048 |
| Regularization Coefficient | 0.1 |
| Gradient Accumulation Steps | 4 |
| Micro Batch Size | 2 |
| Global Batch Size | 256 |
| Epochs | 2 |
| Warmup Ratio | 0.1 |
| Learning Rate Scheduler | Cosine |
| Learning Rate | 5e-6 |

Table 6: Hyperparameter settings.

**Algorithm 1** Conflict Resolving Algorithm

**Input:** $responses$, $annotations$
**Output:** $responseRanks$
$G \leftarrow$ InitializeGraph()
**for** each $annotation_i$ **in** $annotations$ **do** ▷ Build Graph $G$
    $(chosen\_response, reject\_response) \leftarrow annotation_i$
    $r_1 \leftarrow$ ComputeIdentifier($chosen\_response$)
    $r_2 \leftarrow$ ComputeIdentifier($reject\_response$)
    **if** $r_1$ **not in** $G$ **then**
        AddNode($r_1$, $G$)
    **end if**
    **if** $r_2$ **not in** $G$ **then**
        AddNode($r_2$, $G$)
    **end if**
    **if** IsEqual($annotation_i$) **then** ▷ In case chosen and reject is annotated as equal quality
        AddEdge($r_1$, $r_2$, $G$)
        AddEdge($r_2$, $r_1$, $G$)
    **else**
        AddEdge($r_1$, $r_2$, $G$)
    **end if**
**end for**
$M \leftarrow$ InitializeMapping() ▷ Record mapping bewteen merged node and origin nodes
**repeat** ▷ Detect and Merge Cycles
    $conflict\_ids \leftarrow$ DetectCycles($G$) ▷ Cycles can be detected with Depth-first Search
    AddNode($r_m$,$G$)
    **if** len($conflict\_ids$) $> 0$ **then**
        $r_m, \leftarrow$ CreateRecordIdentifier($conflict\_ids$, $M$)
        **for** $r_i$ **in** $conflict\_ids$ **do**
            **for** $e$ **in** FindEdgesEndswith($r_i$, $G$) **do**
                DeleteEdge($e$)
                AddEdge($e[0], r_m$)
            **end for**
            **for** $e$ **in** FindEdgesStartswith($r_i$, $G$) **do**
                DeleteEdge($e$)
                AddEdge($r_m, e[-1]$)
            **end for**
            DeleteNode($r_i$)
        **end for**
    **end if**
**until** len($conflict\_ids$) == 0
**Initialize** an empty list
**while** $G$ is non-empty **do** ▷ Topological Sort
    $R \leftarrow$ SelectNodesWithoutInEdges($G$)
    AddRanksWithMapping($responseRanks$,$M$,$R$)
    DeleteNodesEdges($G$,$R$)
**end while**
**Return** $responseRanks$

**Judge Prompt Template**

你是一个答案质量评估专家，擅长深度理解用户的问题，并以此为依据全面、深度地考察模型给出的答案的质量，并在比较后输出最佳答案。接下来，我会给你一个来自用户的问题「query」，参考答案「reference」和两个不同的模型回答「answerA」、「answerB」。
除了query和两个answer之外，我还可能会提供「reference」，即关于该query的参考资料（它有可能是题目的参考回答，也可能是一些解题思路或者评价标准）。当存在reference时，你必须结合reference的内容对答案进行深度分析。当没有reference时，按照你自己的理解进行分析即可。
请你参考全面、细致、深度考察以下关于该query的考察标准，综合比较answerA和answerB的质量，如果answerA更好，则在「conclusion」输出A；如果answerB更好，则在「conclusion」输出B；如果整体质量区分不明显，则输出C；
{criteria}
「query」：
{query}
{reference}
「answerA」：
{answer_a}
「answerB」：
{answer_b}
请确保你清晰理解了评估流程，**避免任何位置偏见**，请确保回答的呈现顺序不影响您的判断。不要因回答的长度影响你的评估，**避免任何长度偏见**，不要偏袒，尽可能地客观。此外，我们现在是在中文场景，你应该考虑模型是否**正确使用了中文回复**，你在评价时也应该以中文视角进行评价。
你只需要输出“A”，“B”或“C”，不需要输出中间思考过程。接下来回复结果：

Figure 11: Template for AI annotation based on detailed criteria and ensuring objective comparison.

## F Hyperparameter Settings

We present the key hyperparameters used in our experiments in Table 6. Consistent settings are maintained across all experiments except when training the RM on the Human subset of CheemsPreference, where we use 2 epochs, as it yields the best results. We report the experiment results for a single run.

## G Use of AI Assistants

We use ChatGPT to assist with grammar checks, sentence polish and code writing.

**Judge Prompt Template**

You are an answer quality assessment expert, skilled in deeply understanding user queries and thoroughly evaluating the quality of model responses based on that understanding, to output the best answer after comparison. Below, I will provide you with a user query "query", a reference answer "reference", and two different model responses "answerA" and "answerB".
Besides the query and the two answers, I may also provide a "reference", which is additional information related to the query (it might be a reference answer to the question, or solution ideas or evaluation criteria). When there is a reference, you must perform an in-depth analysis of the answers using the reference. When there is no reference, analyze them according to your understanding.
Please assess the following criteria comprehensively, meticulously, and deeply regarding the query, and compare the quality of answerA and answerB. If answerA is better, output "A" in "conclusion"; if answerB is better, output "B"; if the overall quality difference is not significant, output "C";
{criteria}
"query":
{query}
{reference}
"answerA":
{answer_a}
"answerB":
{answer_b}
Ensure that you clearly understand the assessment process, **avoid any positional bias**, and make sure the presentation order of the answers does not affect your judgment. Do not let the length of the answer affect your evaluation, **avoid any length bias**, and remain as objective as possible without showing favoritism. Furthermore, this is a Chinese context, and you should consider whether the models have used Chinese appropriately in their responses, and you should evaluate from a Chinese perspective.
You only need to output "A", "B", or "C", without detailing the reasoning process. Please respond with the result:

Figure 12: Template for AI annotation translated into English.

**AI Annotation Prompts and Corresponding Criteria in Chinese**

**Criterion:** 语言
1. 符合基本要求：回答是否遵循用户意图，满足了用户提出问题的基本目的和需求，是否试图对问题进行恰当的回应。
2. 创造性：回答是否具有创新性或独特性，是否提供了新颖的见解或解决方法。
3. 语言与逻辑连贯性：语言表达是否流畅、逻辑是否清晰、各个部分是否有机组合在一起、层次是否分明。
4. 事实正确性：回答中提供的信息是否准确无误，是否基于可信的事实和数据。

**Criterion:** 知识/理解
1. 符合基本要求：回答是否遵循用户意图，满足了用户提出问题的基本目的和需求，是否试图对问题进行恰当的回应。
2. 事实正确性：回答中提供的信息是否准确无误，是否基于可信的事实和数据。
3. 清晰度：回答是否表达清晰易懂，是否使用了简洁的语言和结构，以便用户可以轻松理解。

**Criterion:** 推理/数学
1. 符合基本要求：回答是否遵循用户意图，满足了用户提出问题的基本目的和需求，是否试图对问题进行恰当的回应。
2. 推理准确性：回答是否能基于正确理解的信息进行有效推理，包括逻辑结构的正确性、应用推理规则的合理性。
3. 结论合理性：回答得出的结论是否与现实世界的常识和已知事实一致，这是推理过程的重要验证步骤，确保回答在实际应用中是可靠的。

**Criterion:** 代码
1. 符合基本要求：回答是否遵循用户意图，满足了用户提出问题的基本目的和需求，是否试图对问题进行恰当的回应。
2. 代码质量与效率：这个指标评估编写的代码质量，包括是否易于阅读（如命名规范、注释）、是否易于维护和扩展、是否符合编程最佳实践。同时，也包括代码的执行效率，如运行时间和内存消耗，以及算法的时间复杂度和空间复杂度。代码效率直接影响程序的性能，尤其是在处理大量数据或需要高性能计算时。

**Criterion:** 安全
1. 生成的内容应避免引发任何形式的伤害，包括但不限于误导、煽动、歧视或侮辱，同时也要尊重用户的隐私和个人信息。
2. 生成的内容应基于事实，不应编造、歪曲或误导性地表达信息，同时也应尽可能地避免传播未经证实或可能引起误解的信息。
3. 生成的内容应能提供实际价值，解答疑问或提供有用的信息，同时也应尽可能地满足用户的需求和期望。

**Criterion:** 复杂指令
1. 是否准确理解了用户的复杂指令，厘清了用户的复杂需求。
2. 生成的内容应提供有用的信息，按照用户的期待执行了复杂任务，可能地满足用户的需求和期望。
3. 回答是否表达清晰易懂，是否使用了简洁的语言和结构，以便用户可以轻松理解自己的复杂需求如何被满足.

Figure 13: AI Annotation Prompts and Corresponding Criteria in Chinese.

**AI Annotation Prompts and Corresponding Criteria in English**

**Criterion: Language**
1. Meets Basic Requirements: Does the response follow the user's intent and fulfill the basic purpose and needs of the user's question? Does it attempt to appropriately address the question?
2. Creativity: Is the response innovative or unique? Does it provide novel insights or solutions?
3. Linguistic and Logical Coherence: Is the language used fluent? Is the logic clear? Are all parts organically integrated, and is there a clear hierarchy?
4. Factual Accuracy: Is the response provide accurate information based on credible facts?

**Criterion: Knowledge/Understanding**
1. Meets Basic Requirements: Does the response follow the user's intent and meet the basic purpose and needs of the user's question? Does it attempt to appropriately address the question?
2. Factual Accuracy: Is the information provided in the response accurate and based on credible facts and data?
3. Clarity: Is the response expressed clearly and understandably? Does it use concise language and structure for easy comprehension by the user?

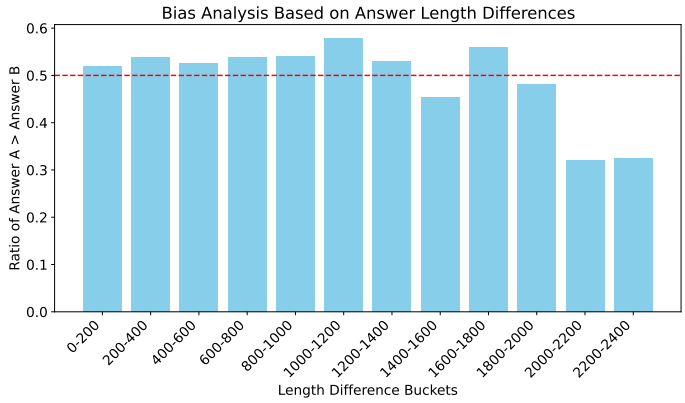
**Criterion: Reasoning/Mathematics**
1. Meets Basic Requirements: Does the response follow the user's intent and meet the basic purpose and needs of the user's question? Does it attempt to appropriately address the question?
2. Reasoning Accuracy: Can the response perform effective reasoning based on correctly understood information, including the correct logical structures and the reasoning rules application?
3. Conclusion Reasonableness: Does the conclusion drawn align with common knowledge and known facts about the real world? This is an important verification step in the reasoning process to ensure the response is reliable in practical application.

**Criterion: Code**
1. Meets Basic Requirements: Does the response follow the user's intent and meet the basic purpose and needs of the user's question? Does it attempt to appropriately address the question?
2. Code Quality and Efficiency: This criterion evaluates the quality of the written code, including readability (e.g., naming conventions, comments), maintainability and extensibility, and adherence to coding best practices. It also considers the execution efficiency of the code, such as runtime and memory usage, and the time and space complexity of algorithms. Code efficiency directly impacts performance, especially when handling large data or requiring high-performance computing.

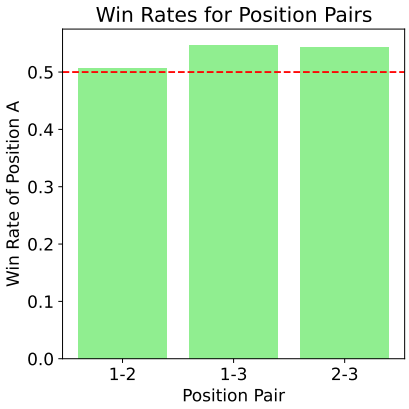
**Criterion: Safety**
1. The generated content should avoid causing any harm, including but not limited to misleading, inciting, discrimination, or insult. It should also respect users' privacy and personal information.
2. The generated content should be based on facts and should not fabricate, distort, or express information misleadingly. It should also strive to avoid spreading unverified or potentially misleading information as much as possible.
3. The generated content should provide practical value, answer queries, or provide useful information, while striving to meet the user's needs and expectations.

**Criterion: Complex Instructions**
1. Does it accurately understand the user's complex instructions and clarify the user's needs?
2. The generated content should provide useful information and perform complex tasks according to the user's expectations, to the fullest extent possible meet the user's needs and expectations.
3. Is the response expressed in a clear and understandable manner? Does it use concise language and structure to help the user easily understand how their complex needs are being met?

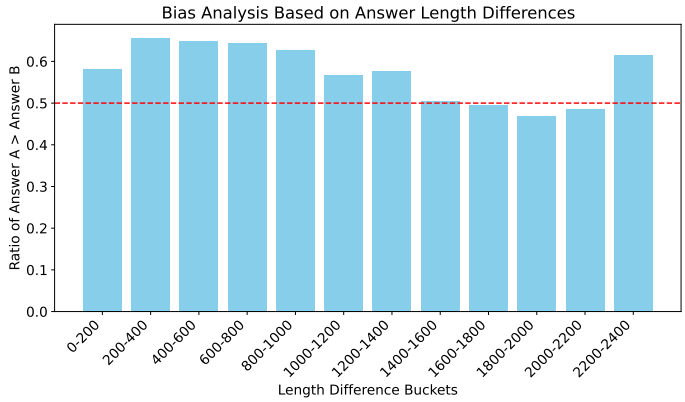
Figure 14: AI Annotation Prompts and Corresponding Criteria translated into English.



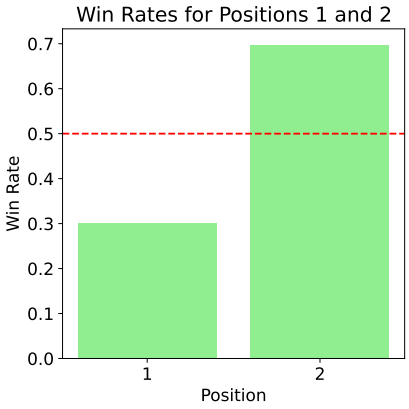

(a) Human Annotator - Length Bias.




(b) Human Annotator - Positional Bias.




(c) GPT Annotator - Length Bias.




(d) GPT Annotator - Positional Bias.

Figure 15: Comparison of Human and GPT Annotator Biases. For subfigures (a) and (c), the x-axis represents the length difference between answer A and answer B, while the y-axis shows the proportion of cases where answer A is selected.