WizardArena: Post-training Large Language Models via Simulated Offline Chatbot Arena

Haipeng Luo[1*] **Qingfeng Sun**[2*] **Can Xu**[2†] **Pu Zhao**[2]

Qingwei Lin[2] **Jianguang Lou**[2] **Shifeng Chen**[3 †] **Yansong Tang**[1 †] **Weizhu Chen**[2]

[1]Shenzhen International Graduate School, Tsinghua University [2]Microsoft Corporation
[3]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
{luohp24@mails., tang.yansong@sz.}tsinghua.edu.cn, {shifeng.chen}@siat.ac.cn
{qins,caxu,puzhao,qlin,jlou,wzchen}@microsoft.com

Abstract

Recent work demonstrates that, post-training large language models with open-domain instruction following data have achieved colossal success. Simultaneously, human Chatbot Arena has emerged as one of the most reasonable benchmarks for model evaluation and developmental guidance. However, the processes of manually curating high-quality training data and utilizing online human evaluation platforms are both expensive and limited. To mitigate the manual and temporal costs associated with post-training, this paper introduces a Simulated Chatbot Arena named *WizardArena*, which is fully based on and powered by open-source LLMs. For evaluation scenario, *WizardArena* can efficiently predict accurate performance rankings among different models based on offline test set. For the training scenario, we propose *Arena Learning*, an innovative offline strategy that simulates iterative arena battles among various state-of-the-art models on a large scale of instruction data using AI-driven annotations to evaluate and leverage battle results, thus continuously enhancing the weaknesses of the target model through both supervised fine-tuning and reinforcement learning. Experimental results demonstrate that our WizardArena aligns closely with the online human arena rankings, and our models, trained on extensive offline battle data through Arena Learning, demonstrate marked improvements in performance across the SFT, DPO, and PPO stages.

1 Introduction

In recent years, the field of natural language processing (NLP) has witnessed a remarkable transformation, driven by the rapid advancements in large language models (LLMs). These models, trained on vast amounts of text data, have demonstrated an exceptional ability to understand, generate, and interact with human language in a wide range of tasks [1–3]. One of the most exciting applications of LLMs has been in the realm of conversational AI [4–9], where they have been utilized to create powerful chatbots capable of engaging in naturalistic dialogues. One of the key factors contributing to the success of LLM-powered chatbots is the ability to leverage large-scale high-quality instruction following data for effective post-training [10–14]. By exposing these models to a diverse range of conversational tasks and instructional scenarios, researchers have been able to imbue them with a deep understanding of how to effectively communicate and assist humans.

[*] Equal contributions. Work done during the internship of Luo at Microsoft.
[†] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3543

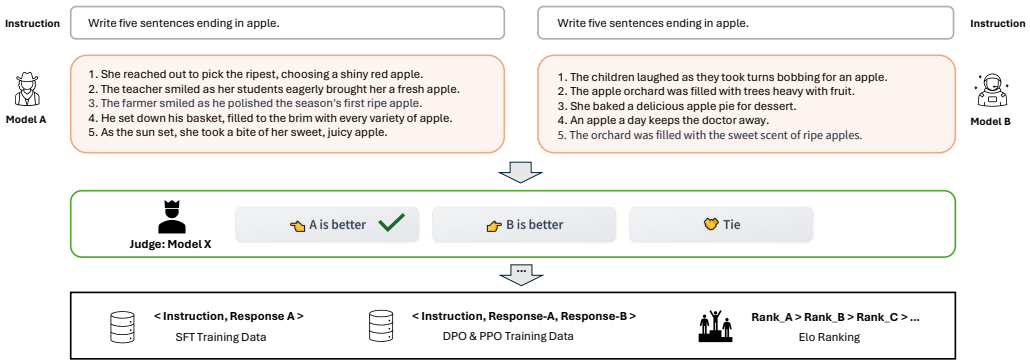

Figure 1: Overview of Running Example.

In this context, the emergence of the Chatbot Arena [15, 16] has been a significant development. This is a platform that facilitates the assessment and comparison of different chatbot models by pitting them against each other in a series of conversational challenges and rank with Elo rating system [17]. By leveraging a diverse set of human evaluators, the Chatbot Arena provides a more robust and comprehensive evaluation of chatbot performance, going beyond the limitations of traditional benchmarking approaches. At the same time, they also opened up some real direct chat and battle preferences data [18], which have been proven to be valuable resources for model post-training and developmental guidance [19]. However, the manual nature of the human-based evaluation process poses its own set of challenges: Manually orchestrating and coordinating the interactions between chatbots and human evaluators can be time-consuming and resource-intensive, limiting the scale and frequency of evaluation and training data opensource cycles. On the other hand, due to their priority limitations [20], most models are unable to participate in arena evaluations, making it impossible to directly guide the development of the target model on it. So, the need for a more efficient and scalable arena-based pipeline to chatbot post-training and evaluation has become increasingly pressing.

To address these challenges, this paper introduces a novel approach called WizardArena, a simulated offline chatbot arena that is fully based on and powered by AI LLMs without human evaluators. The primary objective of WizardArena is to mitigate the manual and temporal costs associated with post-training LLMs while retaining the benefits of arena-based evaluation and training. As the running example shown in the Figure 1, the key is that WizardArena can efficiently predict accurate performance rankings among different battle models based on a powerful "judge model", which could automatically imitate the manner of human annotators in judging a responses pair of two models and provide rankings, scores, and explanation.

In the training scenario, We innovatively propose *Arena Learning* strategy to simulate arena battles among target model (referred to as WizardLM-β) and various state-of-the-art models on a large scale of instruction data. These synthetic battle results are then used to enhance WizardLM-β through some training strategies, including supervised fine-tuning (SFT), direct preference optimization (DPO) [21], and proximal policy optimization (PPO) [22], enabling it to learn from the strengths and weaknesses of other models. Furthermore, *Arena Learning* introduces an iterative battle and training process, where the WizardLM-β is continuously updated and re-evaluated against SOTA models. This process allows for the WizardLM-β to iteratively improve and adapt to the evolving landscape of the arena, ensuring that it remains competitive and up-to-date with the latest advancements in the field.

In the evaluation scenario, we firstly contribute a carefully prepared offline testset, it effectively balances the diversity and complexity of evaluation. By automating the pair judgement process with "judge model", WizardArena significantly reducing the associated costs and priority limitations, and could produce the Elo rankings and detailed win/loss/tie statistics.

The experimental results demonstrate that the Elo rankings produced by WizardArena align closely with the LMSys Chatbot Arena. This finding validates the effectiveness of WizardArena as a reliable and cost-effective alternative to human-based evaluation platforms. Moreover, the models trained on the extensive battle data generated by *Arena Learning* exhibit significant performance improvements during the SFT, DPO, and PPO stages. In three iterative loops, our model can also scale up with more training data and achieve better performance. These results highlight the value and power

of *Arena Learning* in post-training, which leverages the collective knowledge and capabilities of multiple models to drive the WizardLM-β's performance to new heights. Our main contributions are as follows:

- We introduce *Arena Learning*, a novel AI powered method which help us build an efficient data flywheel for large language models post-training by simulating offline chatbot arena, which leverages AI annotator to mitigate the manual and temporal costs.
- We contribute a carefully prepared offline testset of AI-based *WizardArena*, and demonstrate the highly consistent in accurately predicting Elo rankings among different LLMs compared to human-based LMSys Chatbot Arena.
- Experimental results demonstrate the effectiveness of *Arena Learning* in producing large-scale synthetic data to continuously improve WizardLM-β, through various training strategies including SFT, DPO, and PPO.

2 Related Works

2.1 LLM Benchmarks

Large Language Models (LLMs) have transformed the way people interact with computing systems and are extensively used in everyday life and work [23]. The existing benchmarks [24–26] are mainly divided into two categories: 1) Specialized tasks. Knowledge and Capability: MMLU [27], CMMLU [28], and C-Eval [29]; Reasoning: ARC [30], HellaSwag [31], PIQA[32], GSM8k [33], MATH [34]; Programming: HumanEval [35], MBPP [36], LiveCodeBench [37]; Safety and Truthfulness: ToxicChat [38], OLID [39], BIG-Bench [40], TruthfulQA [41]. They focus on assessing LLM performance in specific areas. 2) General tasks: like MT-Bench [15, 42] and AlpacaEval [43], encompass categories such as writing, role-playing, and mathematics, highlighting the models' comprehensive abilities and multi-turn dialogue performance.

Real-world benchmarks, (i.e., LMSYS ChatBot Arena [44] and Allenai WildBench [45]) use anonymous battles, ELO [17, 46] rankings, and human judgments, but have time delay and often do not timely reflect the models' true performance and require large time and human labor intensive. Additionally, most models overfit on leaderboards like MT-Bench [15], OpenLLM leaderboard [47, 48], showing inconsistent performance with real-world ChatBot scenarios and low differentiation among models. Therefore, we have developed Simulated Offline WizardArena, which not only effectively differentiates model performance but also aligns closely with the online live ChatBot Arena [44], making it suitable for selecting the optimal models while significantly enhancing model post-training through battle data.

2.2 Large Language Models

LLMs have made significant strides in Natural Language Processing (NLP), serving as a versatile foundation for numerous applications [23, 49, 50]. These models, which often contain hundreds of billions of parameters, are trained on expansive text datasets. Notable examples include OpenAI's GPT-3 and GPT-4 [4, 51], Anthropic's Claude [52], Google's PaLM [53, 54], Gemini [6], and DeepMind's Chinchilla [55]. The AI field has recently seen a surge in open-source LLMs, providing public access to model codes and parameters. Notable releases include BigScience's BLOOM [56], Mistral AI's Mistral [57], Meta's Llama family [3, 58] and GAL [59], Tsinghua University's ChatGLM [60], TII's Falcon [61] and Yi [62]. New entries such as Baichuan, Qwen [7], DeepSeek [63], and Llemma [64] have also emerged. Presently, models like Alpaca [11], Vicuna [10], Guanaco [65], Orca [66], OpenChat [13], Tulu2 [67], WizardLM [12], and Zephyr [68] are being developed through supervised fine-tuning based on Llama [3, 58] and Mistral [57].

The alignment performance of Large Language Models (LLMs) is significantly influenced by the quality of Supervised Fine-Tuning (SFT) data, which encompasses task difficulty [66], query complexity [12, 69], semantic diversity [11, 14], and sample size [70]. For instance, [11] generates diverse queries through self-instruct [71] methods, while [12] enhances model alignment by increasing query complexity. [66] boosts NLP task performance by optimizing FLAN [72] queries and responses with specialized LLMs, and [14] has introduced UltraChat. To select data efficiently, some strategies like IFD [73], INSTAG [74], DEITA [75], MODS [76], and ALPAGASUS [77] are

 https://doi.org/10.52202/079017-3543

Figure 2: Overview of Arena Learning post-training data flywheel and WizardArena evaluation.

adopted. [66] employs ChatGPT to label instructional data, ensuring both diversity and complexity. Here, we select training data using the judge pair method with different models.

To better adapt to preferences beyond SFT, models are trained with feedback-based methods like RLHF and RLAIF [2, 52, 58, 78, 79], employing Proximal Policy Optimization (PPO) [80] to align with model preferences. RLEIF [81] combines instruction evolution and reinforcement learning to enhance the mathematical reasoning capabilities of the model. Due to RLHF's complexity and instability, simpler alternatives like DPO [21], RRHF [82], KTO [83], IPO [84], sDPO [85], and ORPO [86] are utilized. DPO [21] merges reward modeling with preference learning, RRHF [82] uses ranking loss to prioritize preferred answers, and KTO [83] operates without needing paired preference datasets. In this paper, we use DPO and PPO for model post-training.

3 Approach

In this section, we elaborate on the details of the proposed *WizardArena* and *Arena Learning* strategy. As illustrated in Figure 2, the pipeline mainly contains three components: Offline Pair-size LLM Battle Arena, Model Evaluation, and Model Training.

3.1 ChatBot Arena and Elo Ranking

The Chatbot Arena is a pioneering platform that has revolutionized the way chatbot models are evaluated and compared. It facilitates the assessment of different chatbot models by pitting them against each other in a series of conversational challenges. At the core of this Arena lies the concept of Elo rankings, a widely adopted rating system originally devised for chess players. Elo rankings [17] are used to quantify the relative performance of chatbot models based on a series of head-to-head battles. Each model is initially assigned an arbitrary Elo rating, which is then updated after every battle based on the outcome (win, loss, or tie) and the rating difference between the competing models. If a higher-rated model defeats a lower-rated one, its Elo rating increases slightly, while the loser's rating decreases by a corresponding amount.

3.2 Using a Powerful LLM as Judge to Simulate Human Annotators

At the core of the simulated arena battles in WizardArena lies a powerful LLM that serves as the "judge model". This judge model is specifically prompted and adjusted by us on a diverse range of conversational pair data, enabling it to assess the quality, relevance, and appropriateness of the models' responses objectively and consistently. The judge model's role is to analyze and compare the responses provided by the competing models for each instruction or conversational sample. It considers various factors, such as coherence, factual accuracy, context-awareness, and overall quality, to determine whether one response is superior to the other or if they are of comparable quality (a tie). We show the details of its judgement prompt in Appendix A.

3.3 Evaluation LLMs with WizardArena

3.3.1 Constructing the Offline Test Set

To accurately evaluate the performance of chatbot models and predict their Elo rankings, WizardArena relies on a carefully curated offline mixed test set, which is designed to strike a balance between diversity and complexity, ensuring a comprehensive assessment of the models' capabilities across a wide range of conversational scenarios. This test set consists of the following two subsets:

Diverse Subset The diverse subset of the test set is constructed to capture a broad range of topics, styles, and conversational contexts. To achieve this, we employs text clustering techniques on a large corpus of instructions and conversational data. The clustering process begins by representing all the instructions in a conversation as a high-dimensional vector using state-of-the-art embedding models. These vectors capture the semantic and contextual information within the text, enabling the clustering algorithm to group similar samples together. Once the clustering is complete, we selects a representative sample from each cluster, ensuring that the diverse subset of the test set capture a broad range of scenarios. This approach helps to mitigate potential biases or blindspots that may arise from relying solely on simply random sampling.

Hard Subset This subset is specifically designed to challenge the capabilities of even the most advanced chatbot models. To construct this subset, we leverages the power of LLMs to predict the difficulty level of each instruction. We then selects the top-ranking samples according to the predicted difficulty scores, ensuring that the hard subset of the test set comprises the most challenging and complex scenarios. This data serves as a rigorous benchmark for evaluating the robustness and capability of chatbot models in handling intricate and nuanced conversational tasks.

3.3.2 Simulating Offline Battling and Ranking Models on Test Set

With the above "judge" model and the offline test set in place, WizardArena proceeds to evaluate the performance of various chatbot models through a series of pair-wise battles. The outcomes of the battles are then used to compute the Elo rankings of the participating chatbot models. WizardArena adopts the same Elo rating system used in LMSys Chatbot Arena, which has proven effective in ranking players or entities based on their head-to-head performance.

3.4 Iterative Training LLMs through *Arena Learning*

3.4.1 Collecting Large-Scale Instruction Data

To facilitate leveraging the simulated arena battles among models to train WizardLM-β, *Arena Learning* relies on a large-scale corpus of conversational data D. The data collection process involves several stages of filtering, cleaning, and deduplication to ensure the quality and diversity of the instruction data. The simulated arena battle outcomes are then used to generate training data for the WizardLM-β, tailored to different training strategies: supervised fine-tuning (SFT), direct preference optimization (DPO), and proximal policy optimization (PPO). We split the data equally into some parts $D = \{D_0, D_1, D_2, ..., D_N\}$ for following iterative training and updates respectively.

3.4.2 Iterative Battle and Model Updating

Arena Learning employs an iterative process for training and improving the WizardLM-β. After each round of simulated arena battles and training data generation, the WizardLM-β is updated using the appropriate training strategies (SFT, DPO, and/or PPO). This updated model is then re-introduced into the arena, where it battles against the other SOTA models once again. This iterative process allows the WizardLM-β to continuously improve and adapt to the evolving landscape of the arena. As the model becomes stronger, the simulated battles become more challenging, forcing the WizardLM-β to push its boundaries and learn from the latest strategies and capabilities exhibited by the other models. Additionally, the iterative nature of *Arena Learning* enables the researchers to monitor the progress and performance of the WizardLM-β over time, providing valuable insights into the effectiveness of the different training strategies and potential areas for further improvement or refinement.

The following is the first iterative loop: For SFT, we first train the initial version of WizardLM-β with D_0, then select some state-of-the-art LLMs which ranking top on WizardArena testset, following we battle WizardLM-β with them on D_1 , and focus on extracting instances where the WizardLM-β's

 https://doi.org/10.52202/079017-3543

response is considered inferior to the winning model's response, as determined by the judge model. These instances are collected, and the winning model's response is used as the target output for fine-tuning the next WizardLM-β-SFT model. For DPO, we use WizardLM-β-SFT to battle with SOTA LLMs on D_2, and then we treat win and loss responses as the < choice, reject > pairs to training the WizardLM-β-DPO model. For PPO, we leverage the same battle process on D_3 to obtain the < choice, reject > pairs to train the reward model and WizardLM-β-PPO.

In the second training iteration, we select the latest best WizardLM-β on the WizardArena test set as the initial model, and adopt above battles on D_4, D_5, and D_6 to get the training data of next version of SFT, DPO, and PPO models respectively. We will stop the iteration when we find the model can't achieve significantly better performance than previous iteration.

4 Experiments

4.1 Experimental Setup

Source Data. We collected some instructions from open available datasets (i.e., Alpaca [11], FLAN [72], LMSYS-Chat-1M [87], OpenOrca [88], WizardLM [12]), and optimized them using the following steps: first, we filtered out all illegal conversations; second, we removed conversations with instruction lengths of less than 10; third, we eliminated duplicate instructions with prefixes of 10; next, we employed the MinHashLSH technique [89] (with a threshold of 0.4 and 128 permutation functions) for data deduplication; subsequently, we excluded instructions from the top 5 matches in semantic similarity with benchmarks (i.e., WizardArena, LMSYS-hard [90], MT-Bench [15], AlpacaEval [43], OpenLLM Leaderboard [27, 31, 41, 47, 91]) to prevent data leakage. Finally, we removed all non-English instructions. After completing these steps, we obtained the refined 276K dataset D.

Offline Diverse & Hard WizardArena test set. Firstly, we processed the source data using K-Means clustering into 500 categories. From each category, we randomly selected two samples to construct 1,000 diversity samples, named as the Offline-Diverse WizardArena. Additionally, 20 samples from each category were selected at random to form a data set of 10,000 entries, we then used GPT-4-1106-preview to rate each instruction on a difficulty scale from 0 to 10 in descending order, and selected the top 1,000 entries to create the hard test set, denoted as the Offline-Hard WizardArena. Detailed scoring prompt is provided in Appendix B. The Offline-Mix WizardArena combines the Diverse and Hard test sets in 2,000 samples. Different from Arena-Hard-v1.0 [90], which mainly focuses on single-turn dialogue data, WizardArena-Mix incorporates multi-turn dialogue data.

LLM Battle. We selected 15 popular models from the LMSYS ChatBot Arena and conducted pairwise battles in the Offline-Mix WizardArena. Llama3-70B-Chat [58] served as the "judge" model, with the higher-scoring model declared the winner. To maintain consistency, the detailed judgement prompt is sourced from [15, 92] provided in Appendix A. Following LMSYS Chatbot Arena, we adopt the Bradley-Terry model [93] to calculate the final scores for each model. To mitigate potential position bias, we used a two-game setup, swapping the models between the first and second positions for each instance [92]. We use multiple bootstraps (i.e., 100), and select the median as the model's ELO score. The 95% CI is determined from the 2.5% to 97.5% range of confidence interval scores.

Training Data. 1) we random sample 10k ShareGPT data to train a initial model WizardLM-β-I_0. 2) we randomly split the D into nine slices, each D_i contains around 30K multi-turn conversations. 3) In the first iteration I_1, we inference three SOTA models Command R+, Qwen1.5-72B-chat [94], and OpenChat-3.5 [13] as reference models for battle and We also inference our WizardLM-β-I_0 on D_1. 4) We employ the judge model to judge between WizardLM-β-I_0 and each reference model. 5) The winning reference model's responses (threshold score > 1.0 for maintaining the distinction) are used as the target output to train WizardLM-SFT-β-I_1. 6) Immediately, we use this as initial model and re-battle with three SOTA models on D_2 and D_3 to produce the training data of WizardLM-DPO-β-I_1 and WizardLM-PPO-β-I_1 respectively. The best model from I_1 will be the initial model of second iteration I_2, and the third one I_3 also operates the same way. Finally, we obtain 56.5K (D_0 10K, D_1 20K, D_4 14K, D_7 12.5K) data for SFT, 57.3K (D_2 20.4K, D_5 19.3K, D_8 17.6K) data for DPO, 57.3K (D_3 20.4K, D_6 19.3K, D_9 17.6K) data for PPO reward model, and 90k for PPO.

Implementation Details. We apply our method to the Mistral-7B [57] for post-training, use Llama3-70B-Chat [58] as judge models in WizardArena. In supervised fine-tuning, we trained three epochs

Table 1: The ELO rankings results of 22 models on LMSYS ChatBot Arena EN, MT-Bench, Offline-Diverse, Offline-Hard, and Offline-Mix (Diverse & Hard).

Model	LMSYS-ChatBot Arena-ELO-EN (95% CI)	WizardArena Diverse-ELO (95% CI)	WizardArena Hard-ELO (95% CI)	WizardArena Mix-ELO (95% CI)	MT-bench
Command R+ [97]	1163 (+3/-5)	1353 (+9/-6)	1329 (+8/-6)	1340 (+6/-4)	8.20
Qwen1.5-72B-Chat [94]	1137 (+3/-4)	1334 (+9/-7)	1314 (+7/-5)	1324 (+6/-5)	8.61
Qwen1.5-32B-Chat [94]	1115 (+5/-7)	1299 (+7/-8)	1278 (+6/-8)	1288 (+6/-4)	8.30
Wizard-β-PPO-I_3	-	1283 (+5/-6)	1268 (+7/-5)	1274 (+6/-7)	7.81
Wizard-β-DPO-PPO-I_1	-	1227 (+7/-8)	1208 (+8/-6)	1219 (+7/-5)	7.40
WizardLM-β-PPO-I_1	-	1211 (+7/-8)	1199 (+8/-6)	1205 (+4/-5)	7.29
WizardLM-β-DPO-I_1	-	1203 (+7/-6)	1193 (+8/-8)	1198 (+3/-4)	7.35
WizardLM-70B-v1.0 [12]	1099 (+7/-8)	1187 (+6/-7)	1165 (+6/-5)	1172 (+5/-5)	7.71
Tulu-2-DPO-70B [67]	1093 (+8/-10)	1150 (+7/-6)	1183 (+7/-6)	1161 (+4/-6)	7.89
Llama-2-70B-Chat [58]	1091 (+5/-5)	1100 (+5/-5)	1100 (+4/-6)	1100 (+2/-3)	6.86
Vicuna-33B [10]	1088 (+5/-5)	1115 (+5/-7)	1078 (+7/-5)	1094 (+4/-5)	7.12
Nous-Hermes-2-Mixtral-DPO [98]	1079 (+9/-13)	1109 (+8/-6)	1124 (+7/-7)	1116 (+5/-4)	8.33
WizardLM-β-SFT-I_1	-	1065 (+6/-7)	1061 (+6/-6)	1063 (+4/-4)	6.98
OpenChat-3.5 [13]	1066 (+7/-7)	1043 (+7/-5)	1051 (+8/-5)	1048 (+5/-5)	7.80
DeepSeek-LLM-67B-Chat [9]	1066 (+8/-10)	993 (+7/-7)	1010 (+5/-7)	1001 (+6/-5)	8.70
Llama-2-13B-Chat [58]	1060 (+4/-5)	1053 (+5/-6)	1044 (+7/-7)	1047 (+5/-4)	6.65
GPT-3.5-Turbo-0613 [4]	1055 (+6/-6)	957 (+6/-7)	1008 (+7/-7)	984 (+5/-5)	8.32
Zephyr-7b-alpha [68]	1041 (+14/-15)	907 (+7/-6)	969 (+6/-6)	943 (+4/-4)	6.88
Vicuna-13B [10]	1031 (+5/-6)	937 (+6/-7)	926 (+8/-5)	929 (+5/-5)	6.57
Qwen-14B-Chat [94]	1019 (+9/-10)	917 (+7/-7)	934 (+8/-6)	926 (+4/-6)	6.96
Mistral-7B-Instruct-v0.1 [57]	1011 (+7/-7)	884 (+7/-7)	905 (+9/-6)	895 (+4/-5)	6.84
WizardLM-β-SFT-I_0	-	865 (+6/-7)	887 (+6/-7)	875 (+5/-5)	6.41

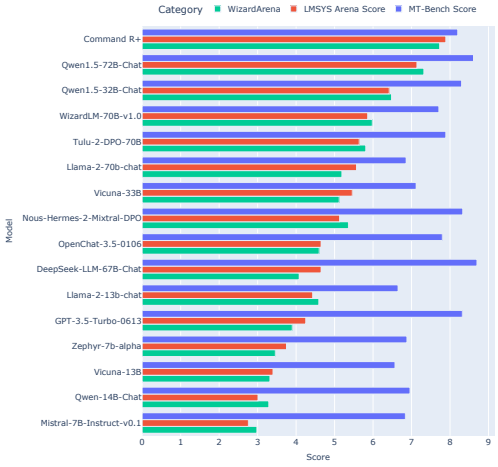

Figure 3: The performance comparison of 15 popular models across MT-Bench, normalized LMSYS ChatBot Arena, and normalized WizardArena.

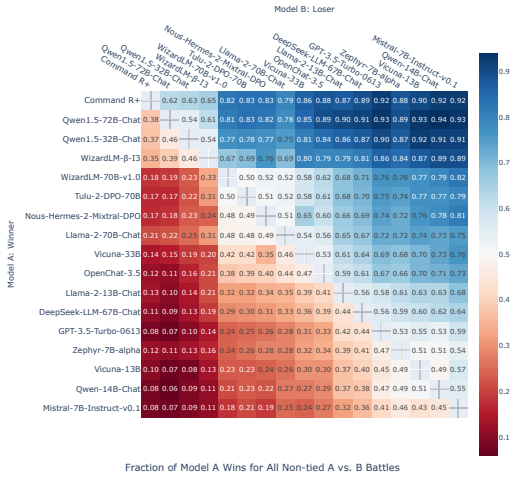

Figure 4: Win rates (w/o tie) of models in WizardArena-Mix. Each model involved in 2k $\times$ 20 battles (#samples num $\times$ #models num).

with a learning rate of 5e-6, a batch size of 128, and a sequence length of 4096. For PPO reward model training, Mistral-7B was trained for one epoch at a learning rate of 1e-6. In PPO training, the learning rate was 1e-7 for one epoch with a KL coefficient of 0.4, and for DPO training, it was 5e-7 for two epochs with a beta of 0.3. We used the DeepSpeed [95] and TRL [96] for SFT and RL.

4.2 Offline WizardArena closely align with the Online LMSYS ChatBot Arena.

Table 1 and Figure 3 presents the rankings for 16 popular models across several evaluation benchmarks: LMSYS ChatBot Arena-EN [44], MT-Bench [15], and three Offline-Diverse, Offline-Hard, and Offline-Mix (Diverse & Hard) of WizardArena. The results reveal that employing the LMSYS ChatBot Arena as the reference benchmark in the real-world scenarios, WizardArena displays the good ranking consistency, however MT-Bench shows the large fluctuations. We find some models that perform well on MT-Bench, such as GPT-3.5-Turbo-0613 [4] and DeepSeek-LLM-67B-Chat [9], rank lower in the LMSYS ChatBot Arena. In addition, there is a significant difference in performance between WizardArena diverse and hard subsets, with Vicuna-33B [10] and Qwen1.5-32B-Chat [94] being more effective in diverse tasks, while Tulu-2-DPO-70B [67] and Nous-Hermes-2-Mixture-DPO [98] achieving better results in hard tasks.

https://doi.org/10.52202/079017-3543

Table 2 illustrates that the Offline WizardArena-Mix significantly outperforms MT-Bench across several consistent metrics, calculated from the evaluation results of the models in Table 1: a 45% higher Spearman Correlation, a 74% increase in Human Agreement with 95% CI, and a 60% improvement in Differentiation with 95% CI. It achieves totally 92.80% consistency with the LMSYS ChatBot Arena , closely matching the 91.57% consistency of Arena-Hard-v1.0 [90]. Details of these metrics are provided in Appendix C. In contrast to MT-Bench and Arena-Hard-v1.0, using proprietary models (i.e. GPT-4) to judge, our approach employs current SOTA open-source models like Llama3-70B-Chat, which has a significantly lower cost. Moreover, the Offline Arena-Mix, which combines Diverse and Hard test sets, achieves 3.27% higher average consistency than only using the Diverse-Arena and 1.8% higher than the LMSys Arena-Hard-v0.1 , indicating greater consistency with the Online LMSYS ChatBot Arena.

Table 2: The consistency of MT-Bench, Arena-Hard-v1.0, and Offline WizardArena compared with the LMSYS ChatBot Arena.

Metrics	MT-Bench	Arena-Hard-v0.1	Offline-Diverse	Offline-Hard	Offline-Mix
Data Size	160	500	1000	1000	2000
Spearman Correlation	52.83%	95.61%	94.03%	95.89%	97.76%
Human Agreement with 95% CI	24.32%	93.70%	95.24%	96.65%	98.98%
Differentiation with 95% CI	21.90%	85.4%	79.32%	80.47%	81.67%
Avg.	33.02%	91.57%	89.53%	91.00%	92.80%

Table 3: Explores data selection strategies for the SFT stage, using 10k samples for each method except for the Original D_1.

Data Selection	Data Size	Offline-Mix Arena-ELO (95% CI)	MT-bench
Original	30k	1027 (+6/-6)	6.59
Random Sample	10k	1012 (+8/-7)	6.49
K-Means Cluster	10k	1024 (+5/-8)	6.63
Instruction Length	10k	1038 (+5/-6)	6.65
IFD [73]	10k	1035 (+7/-7)	6.68
INSTAG [74]	10k	1038 (+5/-7)	6.70
Pair-judge	10k	1049 (+6/-5)	6.76

Table 4: Explore the consistency between Llama3-70B-Chat and GPT-4 as judging models in the Offline-Mix Arena. Using multiple bootstraps (e.g., 100), we select the median as the model's ELO score.

model	LMSYS-ChatBot Arena-ELO-EN (95% CI)	Offline-Mix-Arena-ELO GPT4-judge (95% CI)	Offline-Mix-Arena-ELO Llama3-70B-chat-judge (95% CI)
Command R+ [97]	1163 (+3/-5)	1357 (+4/-3)	1340 (+6/-4)
Qwen1.5-72B-Chat [94]	1137 (+3/-4)	1336 (+5/-5)	1324 (+6/-5)
Qwen1.5-32B-Chat [94]	1115 (+5/-7)	1303 (+4/-6)	1288 (+6/-4)
DeepSeek-LLM-67B-Chat [9]	1066 (+8/-10)	990 (+6/-8)	1001 (+6/-5)
GPT-3.5-Turbo-0613 [4]	1055 (+6/-6)	944 (+7/-5)	984 (+5/-5)
Zephyr-7b-alpha [68]	1041 (+14/-15)	932 (+6/-5)	943 (+4/-4)
Vicuna-13B [10]	1031 (+5/-6)	944 (+6/-5)	929 (+5/-5)
Qwen-14B-Chat [94]	1019 (+9/-10)	924 (+4/-7)	926 (+4/-6)

4.3 Can *Arena Learning* improve models performance via post-training?

Table 1 demonstrates the impact of using the *Arena Learning* method post-train Wizard-β models during the SFT, DPO and PPO stages. In the SFT stage, Wizard-β-I_1, compared to Wizard-β-I_0, achieved a 0.57-point increase on MT-Bench, a 188-point rise in WizardArena-Mix ELO, and advanced 8 places in the ELO rankings. In the RL stage, WizardLM-β-PPO-I_1 outperformed Wizard-β-I_1 by 0.31 points on MT-Bench, increased its ELO score in the WizardArena-Mix by 142 points, and moved up 6 places. WizardLM-β-DPO-I_1 improved by 0.37 points on MT-Bench, and 135 points on WizardArena-Mix ELO, and advanced five places. WizardLM-β-DPO-PPO-I_1 even achieves a 0.42-point increase on MT-Bench, a 156-point rise on WizardArena-Mix ELO and climbed seven places, outperforming both WizardLM-β-DPO-I_1 and WizardLM-β-PPO-I_1 individually. This indicates that continued PPO training based on DPO can further boost model performance. Figure 4 also provides a detailed comparison of the win rates between different models, our final 7B WizardLM-β-I_3 achieve performance that is very close to Qwen1.5-32B-Chat.

Above results highlight that continuous battle with SOTA models with *Arena Learning* and updating weights with new selected data can progressively enhance model capacities compared to its rivals. Hence, *Arena Learning* builds an effective data flywheel and utilizing the *Arena Learning* can significantly improve model performance in post-training.

4.4 Ablation Study

Data Selection strategy. To explore the efficiency of our pair-judge data selection method, we compare it with some widely used data selection strategies. In Table 3, We use 10k samples for each method except for the Original D_1. to ensure a fair comparison, the pair-judge battle method only conducts battles between WizardLM-β-SFT-I_0 and Command R+. The data where WizardLM-β-SFT-I_0 loses are selected, with the corresponding responses taken from Command R+. Additionally, the responses for instructions selected by IFD and INSTAG are also derived from Command R+, rather than the original existing responses.

The results indicate that data selected via the pair-judge method yielded a 22-point improvement in the Offline-Mix Arena ELO over the all original 30k data, surpassed the diversity-based K-Means Cluster method by 25 points, and exceeded the instruction complexity-based INSTAG [74] method by 11 points. On MT-bench, the pair-judge method also demonstrated superior performance, with improvements of 0.17 points over Original, 0.13 points over K-Means Cluster, and 0.06 points over INSTAG. This advantage is attributed to that the pair-judge method focuses on instructions where the base model underperforms, particularly in diverse and complex tasks, effectively addressing the model's weaknesses. These results underscore the effectiveness of the pair-judge method in selecting high-quality data during the SFT phase to target and strengthen the weakness of the base model.

Llama3-Chat Judge and GPT-4 Judge Consistency. In most previous works, people were accustomed to use GPT-4 as a judge for evaluation or generating synthetic data, but the GPT-4 API cost required for large-scale data flywheel is enormous for most research and production scenarios. Therefore, we explore whether it is possible to replace GPT-4 with advanced open source models. Table 4 explores the consistency between Llama3-70B-Chat and GPT-4 as judge models in the Offline-Mix Arena. Using GPT-4 judge's ELO as the reference benchmark, the Spearman correlation coefficient between Llama3-70B-Chat judge and GPT-4 judge is 95.81%, and the Human Agreement with 95% CI is 88.46%. The overall average consistency between the two judge models is 92.14%. Consequently, employing Llama3-70B-Instruct as a cost-effective judge model achieves high consistency with both GPT-4 and LMSYS ChatBot Arena by human judgment, ensuring the reliability of the WizardArena evaluation and post-training with *Arena Learning* in this paper.

Table 5: Explore different alignment strategies for models in SFT and RL stages. We utilize three slices of data for SFT, DPO, and PPO training.

Alignment Strategy	Data Source	Offline-Mix Arena-ELO (95% CI)	MT-bench
SFT	D_1	1063 (+4/-4)	6.98
SFT	$D_1 \cup D_2$	1124 (+6/-4)	7.15
SFT + DPO	$D_1 \cup D_2$	1198 (+3/-4)	7.35
SFT + PPO	$D_1 \cup D_2$	1205 (+4/-5)	7.29
SFT + DPO + PPO	$D_1 \cup D_2 \cup D_3$	1219 (+7/-5)	7.40

Table 6: Explore our model's performance across various benchmarks, including the OpenLLM Leaderboard, GSM8k, AlpacaEval2.0 and MT-Bench.

Model	ARC	Hellaswag	MMLU	TruthfulQA	GSM8k	AlpacaEval2.0	MT-bench	Avg
Qwen1.5-7B-Chat [94]	55.89	78.56	61.70	53.65	13.19	14.70	7.60	40.76
Mistral-7B-Instruct-v0.1 [57]	54.52	75.63	55.38	56.28	14.25	8.43	6.84	38.76
Vicuna-13B-v1.5 [10]	57.08	81.24	56.67	51.51	11.30	10.50	6.57	39.27
Zephyr-7B-alpha [68]	61.01	84.04	61.39	57.9	14.03	10.30	6.88	42.22
Llama-2-13B-Chat [58]	49.04	80.70	54.80	41.86	28.70	8.40	6.65	38.59
OpenChat-3.5 [13]	62.46	83.96	62.89	45.43	25.78	11.37	7.80	42.81
WizardLM-β-SFT-I_0	54.73	72.67	54.43	49.16	25.30	8.24	6.41	38.71
WizardLM-β-SFT-I_1	59.94	83.31	60.62	52.49	43.16	13.71	6.98	45.74
WizardLM-β-DPO-I_1	61.12	84.13	61.97	53.98	45.05	17.08	7.35	47.24
WizardLM-β-PPO-I_1	61.24	84.26	62.31	54.24	45.56	17.23	7.29	47.45

Training strategy. Table 5 explores the impact of different training strategies in the first round during the SFT, DPO, and PPO stages. Iterative application of the pair-judge method consistently boosts SFT model performance, exemplified by the Offline-Arena Mix ELO score rising from 1063 to 1124 and the MT-bench score from 6.98 to 7.15. These outcomes confirm the effectiveness and scalability of the pair-judge approach for SFT data selection. In the RL stage, by continuing the post-training of DPO and PPO on top of SFT, the Offline-Arena Mix ELO score significantly increased by 135 points and 142 points, and MT-bench improved by 0.37 points and 0.31 points. Futhermore, SFT + DPO + PPO showed a modest 0.05-point improvement on MT-bench compared to SFT + DPO, but obviously increased by 21 points on Offline-Arena Mix ELO. These findings suggest that the continuous application of reinforcement learning strategies can further boost the model's intrinsic capabilities. Above results indicate that the data derived from the pair-judge battle method not only significantly enhanced the SFT phase training but also provided high-quality data pairs for the RL phase, continuously improving the training outcomes for DPO and PPO.

Scaling Iterative SFT, DPO, PPO training. Figure 5 explores the iterative training processes of SFT, DPO, and PPO, where I_i represents the i-th iteration. The results highlight that continuous battle with WizardAerna and updating can progressively enhance model performance. Specifically, from SFT-I_0 to DPO-I_3 or PPO-I_3, the WizardArena ELO score increased from 875 to 1274, achieves a huge gain of 399 points, and the MT-Bench score also rises from 6.41 to 7.81, achieves an increase of 1.4 points. These findings underscore the effectiveness and scalability of the *Arena Learning* iterative training method in post-training LLMs.

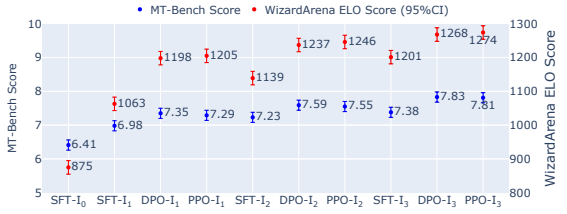

Figure 5: Explore the iterative training processes of SFT, DPO, and PPO. I_i represents the i-th iteration.

https://doi.org/10.52202/079017-3543

Table 7: Explore the quantity of Choose and Reject responses for each battle model across various rounds during the DPO stages.

Stage	Command R+	Qwen1.5-72B-Chat	OpenChat-3.5	WizardLM-β-SFT	Total
DPO-I_1-Choose	9.5k	7.9k	1.4k	1.6k	20.4k
DPO-I_2-Choose	8.8k	7.5k	1.1k	1.9k	19.3k
DPO-I_3-Choose	7.8k	6.6k	0.7k	2.5k	17.6k
DPO-$Total$-Choose	26.1k	22.0k	3.2k	6.0k	57.3k
DPO-I_1-Reject	1.1k	1.6k	9.2k	8.5k	20.4k
DPO-I_2-Reject	0.9k	1.5k	10.5k	6.4k	19.3k
DPO-I_3-Reject	0.9k	1.3k	11.2k	4.2k	17.6k
DPO-$Total$-Reject	2.9k	4.4k	30.9k	19.1k	57.3k

Count of data selected from each battle model during DPO. Table 7 summarizes the sources of Choose and Reject responses during the DPO data construction. Command R+ selected 9.5k, 8.8k, and 7.8k data as Choose responses across three rounds, totaling 26.1k. The corresponding Reject responses were 1.1k, 0.9k, and 0.9k, totaling 2.9k. WizardLM-β-SFT selected 1.6k, 1.9k, and 2.5k Choose responses across three rounds, totaling 6.0k(6.0k vs. 57.3k), with corresponding Reject responses of 8.5k, 6.4k, and 4.2k, totaling 19.1k(19.1k vs. 57.3k). This indicates that as WizardLM-β-SFT improved through iterative training, the number of Choose responses increased, while Reject responses decreased.

Llama3-70B-Chat vs. Human Judge. To reduce time and annotation costs, we randomly selected 200 samples from WizardArena-Mix (100 diverse, 100 challenging). We evaluated four models: WizardLM-β-PPO-I_3 (reference model), OpenChat-3.5, Command R+, and Qwen1.5-72B-Chat (battle models), using Llama3-70B-Chat and professional human annotators , with results shown in Table 8. Llama3-70B-Chat's win rates for WizardLM-β-PPO-I_3 against Command R+, Qwen1.5-72B-Chat, and OpenChat-3.5 were 34.1%, 41.3%, and 79.7%, closely matching human evaluations (31.8%, 37.7%, 82.1%). The high consistency between them further validates Llama3-70B-Chat's reliability and accuracy as a judge model in WizardArena.

Table 8: The win/tie/loss counts of WizardLM-β-PPO-I_3 against Command R+, Qwen1.5-72B-Chat, OpenChat-3.5 evaluated by Llama3 70B Chat Judge and Human Judge.

Models	Win	Tie	Loss
Llama3 70B Chat Judge			
WizardLM-β-PPO-I_3 vs. Command R+	55	39	106
WizardLM-β-PPO-I_3 vs. Qwen1.5-72B-Chat	64	45	91
WizardLM-β-PPO-I_3 vs. OpenChat-3.5	141	23	36
Human Judge			
WizardLM-β-PPO-I_3 vs. Command R+	49	46	105
WizardLM-β-PPO-I_3 vs. Qwen1.5-72B-Chat	60	41	99
WizardLM-β-PPO-I_3 vs. OpenChat-3.5	147	21	32

Performance on more benchmarks. Table 6 showcases our model's performance at the first iteration across various benchmarks, including the OpenLLM Leaderboard, GSM8k, AlpacaEval2.0, and MT-Bench for SFT, DPO and PPO stages. Utilizing the WizardArena method to produce training data has markedly improved model performance in both SFT and RL stages. Specifically, WizardLM-β-SFT-I_1 exceeds WizardLM-β-SFT-I_0 by 7.03 average points. More impressively, WizardLM-β-PPO-I_1 not only surpasses WizardLM-β-SFT-I_0 by 8.74 points but also exceeds WizardLM-β-SFT-I_1 by 1.71 points and outperforms Openchat-3.5 by 4.64 points. Particularly in the reasoning tasks, WizardLM-β-PPO-I_1 shows a 7.88 point increase on MMLU and a significant 20.26 point gain on GSM8k compared to WizardLM-β-SFT-I_0, demonstrating that the our method effectively enhances the model's weaknesses. The detailed scores of WizardLM-β-I_1 SFT, DPO, PPO in the 8 subtasks of MT-Bench refer to the Figure 6.

5 Conclusion

In conclusion, this paper presents *WizardArena*, a simulated offline chatbot arena that utilizes AI LLMs to eliminate the manual and temporal costs associated with post-training LLMs, while preserving the benefits of arena-based evaluation and training. The effectiveness of WizardArena is validated through the high consistency in predicting Elo rankings among different LLMs compared to the human-based LMSys Chatbot Arena. Furthermore, the model trained on synthetic data generated by *Arena Learning* strategy exhibits significant performance improvements across various training strategies. This work showcases the potential of WizardArena as a cost-effective and reliable alternative to human-based evaluation and data production platforms for post-training chatbot models.

Limitations and Broader Impacts. If the judge model fails to accurately imitate human evaluators, the generated rankings and training data may be compromised. Moreover, similar to the other LLMs, our model could also generate potentially unethical or misleading information sometimes.

Acknowledgements

This work was supported in part by Young Elite Scientists Sponsorship Program by CAST (No. 2024QNRC003).

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Long Ouyang, Jeffrey 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 F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

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

[4] OpenAI. Gpt-4 technical report, 2023.

[5] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

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

[7] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenhang Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, K. Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Jian Yang, Shusheng Yang, Shusheng Yang, Bowen Yu, Yu Bowen, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xing Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *ArXiv*, abs/2309.16609, 2023.

[8] BaichuanAI. Baichuan.

[9] DeepseekAI. Deepseek.

[10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[11] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

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

[13] Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*, 2023.

[14] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

[15] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024.

 https://doi.org/10.52202/079017-3543

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

[18] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world llm conversation dataset, 2024.

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

[20] LMSYS. Lmsys chatbot arena: Live and community-driven llm evaluation.

[21] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290, 2023.

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

[23] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[24] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

[25] Jiayin Wang, Fengran Mo, Weizhi Ma, Peijie Sun, Min Zhang, and Jian-Yun Nie. A user-centric benchmark for evaluating large language models. *arXiv preprint arXiv:2404.13940*, 2024.

[26] Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, Deyi Xiong, et al. Evaluating large language models: A comprehensive survey. *arXiv preprint arXiv:2310.19736*, 2023.

[27] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[28] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*, 2023.

[29] Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36, 2024.

[30] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[31] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[32] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press, 2020.

[33] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[34] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[35] 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. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[36] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021.

[37] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

[38] Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation. *arXiv preprint arXiv:2310.17389*, 2023.

[39] Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. Solid: A large-scale semi-supervised dataset for offensive language identification, 2021.

[40] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[41] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

[42] Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, et al. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. *arXiv preprint arXiv:2402.14762*, 2024.

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

[44] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

[45] Bill Yuchen Lin, Khyathi Chandu, Faeze Brahman, Yuntian Deng, Abhilasha Ravichander, Valentina Pyatkin, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild, 2024.

[46] Meriem Boubdir, Edward Kim, Beyza Ermis, Sara Hooker, and Marzieh Fadaee. Elo uncovered: Robustness and best practices in language model evaluation, 2023.

[47] Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, 2023.

[48] Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, September 2021.

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

[50] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

[51] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language

 https://doi.org/10.52202/079017-3543

models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

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

[53] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

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

[55] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *CoRR*, abs/2203.15556, 2022.

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

[57] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

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

[59] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

[60] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

[61] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023.

[62] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

[63] DeepSeek-AI Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wen-Hui Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao,

Jun-Mei Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Min Tang, Bing-Li Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yi Xiong, Hanwei Xu, Ronald X Xu, Yanhong Xu, Dejian Yang, Yu mei You, Shuiping Yu, Xin yuan Yu, Bo Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghu Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. Deepseek llm: Scaling open-source language models with longtermism. *ArXiv*, abs/2401.02954, 2024.

[64] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

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

[66] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023.

[67] Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*, 2023.

[68] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment. *ArXiv*, abs/2310.16944, 2023.

[69] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. *arXiv preprint arXiv:2306.08568*, 2023.

[70] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[72] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR, 2023.

[73] Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*, 2023.

[74] Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, and Chang Zhou. # instag: Instruction tagging for diversity and complexity analysis. *arXiv preprint arXiv:2308.07074*, 2023.

[75] Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*, 2023.

[76] Qianlong Du, Chengqing Zong, and Jiajun Zhang. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653*, 2023.

[77] Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*, 2023.

[78] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

[79] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

 https://doi.org/10.52202/079017-3543

[80] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[81] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

[82] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.

[83] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *ArXiv*, abs/2402.01306, 2024.

[84] Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *ArXiv*, abs/2310.12036, 2023.

[85] Dahyun Kim, Yungi Kim, Wonho Song, Hyeonwoo Kim, Yunsu Kim, Sanghoon Kim, and Chanjun Park. sdpo: Don't use your data all at once. *ArXiv*, abs/2403.19270, 2024.

[86] Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *ArXiv*, abs/2403.07691, 2024.

[87] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Haotong Zhang. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *ArXiv*, abs/2309.11998, 2023.

[88] Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://huggingface.co/Open-Orca/OpenOrca`, 2023.

[89] Anshumali Shrivastava and Ping Li. In defense of minhash over simhash. *ArXiv*, abs/1407.4416, 2014.

[90] Tianle* Li, Wei-Lin* Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From live data to high-quality benchmarks: The arena-hard pipeline, April 2024.

[91] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

[92] Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. *ArXiv*, abs/2305.17926, 2023.

[93] Julien Fageot, Sadegh Farhadkhani, Lê Nguyên Hoang, and Oscar Villemaud. Generalized bradley-terry models for score estimation from paired comparisons. In *AAAI Conference on Artificial Intelligence*, 2023.

[94] Ali. Qwen.

[95] Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase, and Yuxiong He. Deepspeed- inference: Enabling efficient inference of transformer models at unprecedented scale. *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2022.

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

[97] Cohere Inc. Cohere: Large language models for your business.

[98] Teknium, theemozilla, karan4d, and huemin_art. Nous hermes 2 mixtral 8x7b dpo.

A Models Pair-wise Judgement Prompt

Example 1: Models Pair-wise Judgement Prompt

<|The Start of Assistant A's Conversation with User|>

User:
{INSTRUCTION}

Assistant A:
{Assistant A Response}

<|The End of Assistant A's Conversation with User|>

<|The Start of Assistant B's Conversation with User|>

User:
{INSTRUCTION}

Assistant B:
{Assistant B Response}

<|The End of Assistant B's Conversation with User|>

[System]
We would like to request your feedback on the performance of two AI assistants in response to the user question displayed above.
Please rate the helpfulness, relevance, accuracy, level of details of their responses.

Your evaluation should focus on the assistant's answer to the user's last question.
Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance.
Please first provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.
Then, output two lines indicating the scores for Assistant 1 and 2, respectively.

Output with the following format:
Evaluation evidence: <your evluation explanation here>
Score of the Assistant 1: <score>
Score of the Assistant 2: <score>

https://doi.org/10.52202/079017-3543

B GPT-4 Scoring Prompt

Example 2: GPT-4 Scoring Prompt

[System]
We are interested in understanding how well the following input prompts can evaluate an AI assistant's proficiency in problem-solving ability, creativity, or adherence to real-world facts. Your task is to assess each prompt within a conversation based on its potential to gauge the AI's capabilities effectively in these areas.

There are multiple rounds of user prompts in the conversation, please rate each user prompt individually.

Guidelines for Scoring:

• High Score (8-10): Reserved for prompts that are particularly challenging and excellently designed to assess AI proficiency.
• Medium Score (4-7): Given to prompts that have a moderate potential to assess the AI's capabilities.
• Low Score (1-3): Allocated to prompts that are either too easy, ambiguous, or do not adequately assess the AI's capabilities.

our input is a JSON list format, where each element is a dictionary:
- idx: represent which round of user prompts in a conversation.
- value: represent a round of user prompts within a conversation

Your output is in JSON list format, where each element is a dictionary:
- idx: represent which round of user prompts in a conversation.
- value: represent detailed scoring reasons for the round of user prompts within a conversation.
You should carry out the two steps:
1. Assess the Potential: Consider how challenging the prompt is, and how well it can assess an AI's problem-solving skills, creativity, or factual accuracy. Briefly explain your reasoning.
2. Assign a Score: Assign a score on a scale of 1 to 10, with a higher score representing a higher potential to evaluate the AI assistant's proficiency effectively. Use double square brackets to format your scores, like so: [[5]].

Ensure to critically evaluate each prompt and avoid giving high scores to prompts that are ambiguous or too straightforward.

Here is an example.
[### input ###]: [{"idx": 0, "value": "Please show me how to serve a ReactJS app from a simple ExpressJS server. Use typescript."}]

[### output ###]: [{"idx": 0, "value": "1. Assess the Potential: This prompt is a good test of the AI's problem-solving skills and its adherence to real-world facts. The AI would need to provide a step-by-step guide on how to serve a ReactJS app from an ExpressJS server using TypeScript, which requires a good understanding of these technologies. However, it doesn't directly test the AI's creativity.\n2. Assign a Score: This prompt has a high potential to assess the AI's problem-solving skills and adherence to real-world facts, but it doesn't test creativity. Therefore, I would assign a score of [[8]]."}]

It is important to note that the output must be in JSON format and be parsed correctly by the JSON tool.

[### input ###]: {Your_JSON_INSTRUCTION}

[### output ###]:

C Three consistency metrics between two Arenas

To more effectively align the online arena (i.e. LMSYS ChatBot Arena) with real-world human preferences and to enhance differentiation among models, we developed a simulated offline arena. This platform is designed to evaluate the actual performance of the models and to facilitate the selection of optimal model checkpoints. We use several key criteria [90] that define an effective benchmark for evaluating Large Language Models (LLMs) in chatbot applications, aiming to enable meaningful functional comparisons across different models.

- Alignment with Human Preference : The benchmarks should maintain high alignment with real-world human preferences in responses to the diverse and hard instructions, ensuring that the models' outputs meet user expectations.
- Ranking Accuracy: The benchmark should align closely with the reference standard to ensure that the rankings of different models on the leaderboard are reliable and accurate.
- Differentiation: The benchmark should be capable of accurately differentiating the performance of various models by providing confidence intervals with minimal overlap. This feature is crucial to ensure that the more effective models can be reliably distinguished.

We define the alignment of Benchmark A with reference to Benchmark B, for a model pair (m_1, m_2) that B can confidently differentiate, using the following formulation:

The agreement score, $s(m_1, m_2)$, is determined as:

$$s(m_1, m_2) = \begin{cases} 1.0 & \text{if } A \text{ confidently separates } m_1 \text{ from } m_2 \text{ and their ranking aligns with } B \\ -1.0 & \text{if } A \text{ confidently separates } m_1 \text{ from } m_2 \text{ and their ranking conflicts with } B \\ 0.0 & \text{if } A \text{ cannot confidently separate } m_1 \text{ from } m_2 \end{cases}$$

To assess ranking accuracy, we employed Spearman's rank correlation coefficient to analyze the correlation between the two sets of ranking data.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where ρ is the Spearman's rank correlation coefficient, d_i is the difference between the ranks of corresponding variables, and n is the number of observations.

We define the differentiation of models based on their performance scores, which are represented by confidence intervals CI_1 and CI_2 via bootstrapping. If the two confidence intervals do not overlap, then models M_1 and M_2 are considered to be separable.

$$CI_1 \cap CI_2 = \emptyset$$

 https://doi.org/10.52202/079017-3543

D The Radar plot of MT-Bench

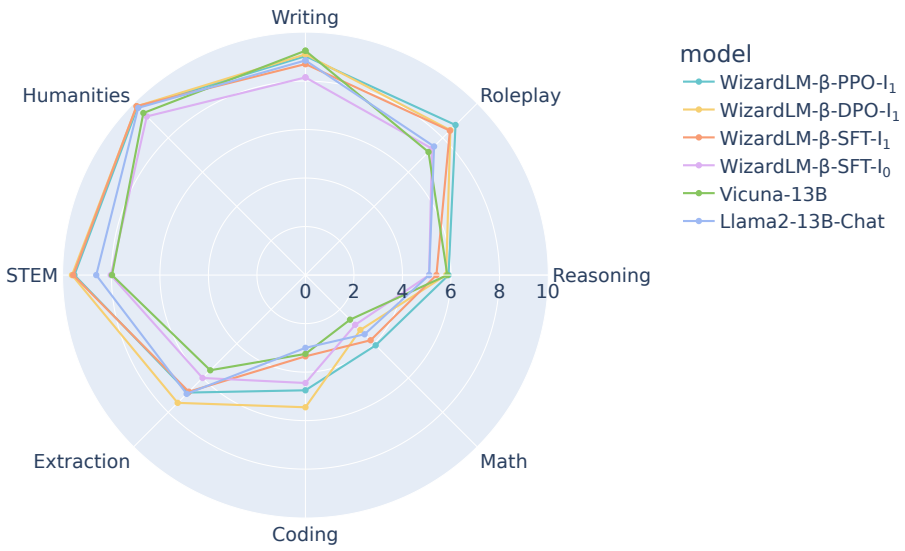

Figure 6: Radar plot showing detailed scores of WizardLM-β-SFT, DPO, PPO at the first iteration in the eight subtasks of MT-Bench.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, please refer to the abstract and introduction of this article for details.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, please refer to the conclusion of this article for details.

https://doi.org/10.52202/079017-3543

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Yes, please refer to the approach of this article for details.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, please refer to the experiments of this article for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Yes, please refer to the appendix source code of this article for details.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

 https://doi.org/10.52202/079017-3543

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, please refer to the experiments of this article for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, please refer to the experiments of this article for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, please refer to the experiments of this article for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, please refer to the experiments of this article for details.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, please refer to the conclusion of this article for details.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Yes, please refer to the conclusion of this article for details.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

https://doi.org/10.52202/079017-3543

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, please refer to the experiments of this article for details.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA] .

    Justification: [NA] .

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer:[NA]

    Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-3543