Arena Learning : Build Data Flywheel for LLMs Post-training via Simulated Chatbot Arena

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

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

[1]Microsoft Corporation
[2]Tsinghua University, [3]SIAT-UCAS

Abstract

Assessing the effectiveness of large language models (LLMs) presents significant challenges. The method of conducting human-annotated battles in an online Chatbot Arena has been recognized as a highly effective evaluative approach. However, this process is hindered by the costliness and time demands of human annotation, complicating the enhancement of LLMs via post-training. In this paper, we introduce "*Arena Learning*", an innovative offline strategy designed to simulate these arena battles. We have developed a comprehensive set of instructions for simulated battles and employ AI-driven annotations to assess battle outcomes, facilitating continuous improvement of the target model through both supervised fine-tuning and reinforcement learning. A crucial aspect of our methodology is ensuring precise evaluations and achieving consistency between offline simulations and online competitions. To this end, we present WizardArena, a pipeline crafted to accurately predict the Elo rankings of various models using a meticulously designed offline test set. Our findings indicate that WizardArena's predictions are closely aligned with those from the online Arena. We apply this novel framework to train our model, WizardLM-β, which demonstrates significant performance enhancements across various metrics. This fully automated training and evaluation pipeline paves the way for ongoing, incremental advancements in various LLMs via post-training. Notably, *Arena Learning* plays a pivotal role in the success of WizardLM-2, and this paper serves both as an exploration of its efficacy and a foundational study for upcoming discussions related to WizardLM-2 and its derivatives.

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 [4–8], 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 [9–13]. 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 HL at Microsoft.

Preprint. Under review.

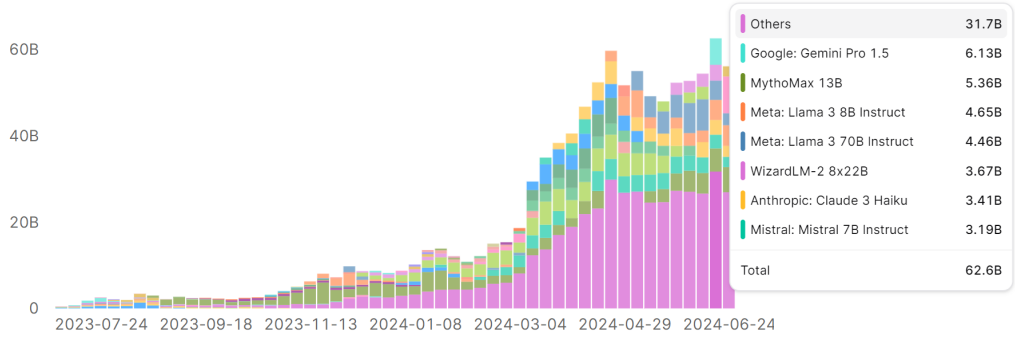

Figure 1: OpenRouter LLM Rankings on processed tokens (https://openrouter.ai/rankings).

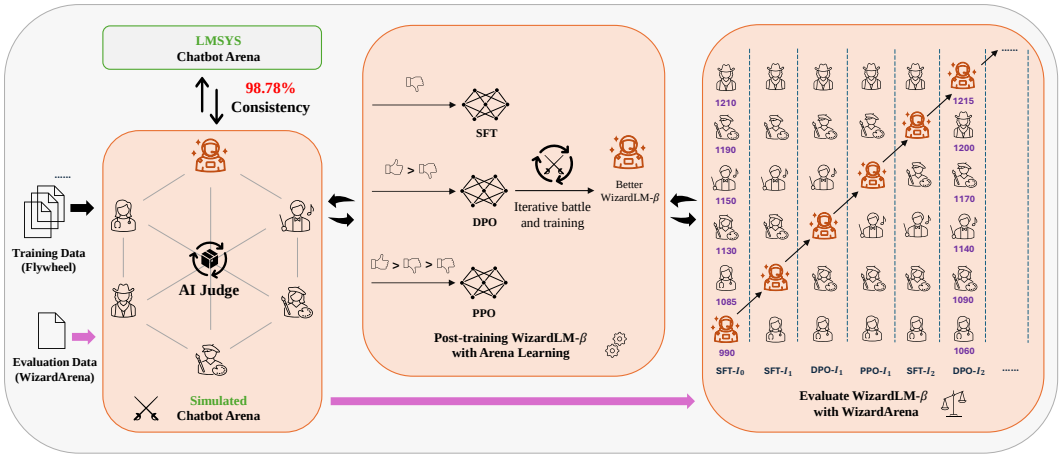

Figure 2: Overview of *Arena Learning* post-training data flywheel and WizardArena evaluation.

With the rapid implementation of various large model applications and the reduction of inference costs, the interest and demand from businesses and consumers in using large language model services have increased rapidly. As shown in the Figure 1, just the OpenRouter platform will process more than 60B tokens every day. At the same time, with the innovation and deepening of application scenarios, this requires those models to continue to evolve to adapt to the user's new intentions and instructions. Therefore, building an efficient data flywheel to continuously collect feedback and improve model capabilities has become a key direction for next generation AI research.

In this context, the emergence of the LMSYS Chatbot Arena [14, 15] 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 [16]. 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, it also opened up some real direct chat and battle preferences data [17], which have been proven to be valuable resources for model post-training and developmental guidance [18]. However, the human-based evaluation process poses its own challenges: Manually orchestrating and waiting 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 [19], most models are unable to participate in arena evaluations, and the community can only obtain 10% of the chat data at most, making it hard to directly and efficiently guide the development of the target model based on this Arena. Therefore, 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 *Arena Learning*, which is a training and evaluation pipeline fully based on and powered by AI LLMs without human evaluators. The primary objective of *Arena Learning* is to build an efficient data flywheel and mitigate the manual and temporal costs associated with post-training LLMs while retaining the benefits of arena-based

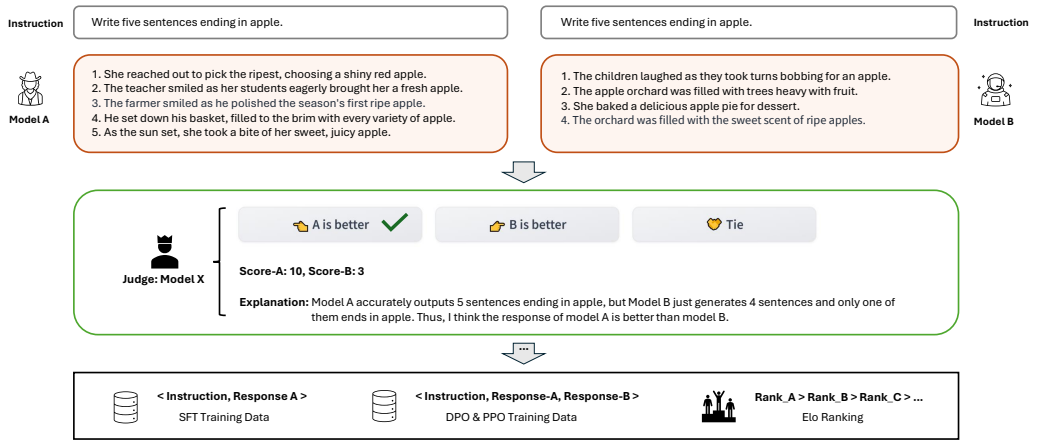

Figure 3: Overview of Running Example: how we use simulated AI-powered pair wise battle arena to produce post-training data and evaluate models.

evaluation and training. As the running example shown in the Figure 3, the key is that *Arena Learning* simulates a offline chatbot arena, and can efficiently predict accurate performance rankings among different arena 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 post-training scenario, as shown in the Figure 2, *Arena Learning* simulates 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) [20], and proximal policy optimization (PPO) [21], enabling it to learn from the strengths and weaknesses of other good 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 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 top-tier competitors in the field.

In the evaluation scenario, we firstly contribute a carefully prepared offline testset - WizardArena, 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 achieve an average consistency of 98.79% with the LMSys Chatbot Arena, outperforming Arena-Hard-v1.0 by 8.58% and MT-Bench by 35.23%. This finding not only validates the effectiveness of WizardArena as a reliable and cost-effective alternative to human-based evaluation platforms, but also further proves the reliability of using the "judge" model to generate a large amount of battle training data in simulated arena. 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 achieve significant improvements in each round compared to the previous one, revealing that *Arena Learning* can scale up to more training data. 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 a new height. 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 - WizardArena, and demonstrate the highly consistent of it 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 flywheel to continuously improve WizardLM-β, through various training strategies including SFT, DPO, and PPO.

2 Approach

In this section, we elaborate on the details of the proposed *Arena Learning*. As illustrated in Figure 2, the closed loop pipeline mainly contains three components: Offline Pair-wise LLM Battle Arena, Iterative Post-training, and Model Evaluation.

2.1 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 [16] 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.

2.2 Using a Powerful LLM as Judge to Simulate Human Annotators

At the core of the simulated arena battles in *Arena Learning* 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 evaluate 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 pair battle models for each conversational sample. Specifically, to assess the response quality of each LLM, we use prompt engineering with the Llama3-70B-Chat model [22]. The inputs are dialogue history, user instruction, and the responses of two LLMs. The outputs consist of scores for each LLM, along with explanations focused on various factors, such as coherence, factual accuracy, context-awareness, and overall quality, to determine whether one response is superior to the other. To mitigate potential position bias [14, 23, 24], we employ a two-game setup, alternating the positions of the two LLMs. Each model receives an overall score on a scale of 1 to 10, where a higher score reflects superior overall performance. Following, we will use this "judge" model in both *Arena Learning* post-training and WizardArena evaluation stages.

2.3 Build a Data Flywheel to Post-train LLMs

2.3.1 Collect 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.

2.3.2 Iterative Battle and Model Evolving

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 training iteration I_1: Before that, we first train the initial version of WizardLM-β-SFT-I_0 with D_0, then select some other state-of-the-art LLMs M which ranking top on WizardArena testset, following we let WizardLM-β-SFT-I_0 as the competitor model, and battle with M on D_1 , and focus on extracting instances where the WizardLM-β's 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-I_1 model. For DPO, we use WizardLM-β-SFT-I_1 as competitor to battle with M on D_2, and then we treat win and loss responses as the < choice, reject > pairs to training the WizardLM-β-DPO-I_1. For PPO, we leverage the same battle process between WizardLM-β-DPO-I_1 and M on D_3 to obtain the < choice, reject > pairs to train the reward model and WizardLM-β-PPO-I_1. In the second training iteration I_2, we select the best WizardLM-β-PPO-I_1 on the WizardArena as the initial competitor model of I_2, and adopt similar process to train next SFT, DPO, and PPO models. Table 1 shows the details of data and models used in each stage.

Table 1: Data and models used in different training stages

New Model	Train From	Competitor Model	Training Data
SFT-I_0	Mistral-Base	-	D_0
SFT-I_1	Mistral-Base	SFT-I_0	$D_0 \cup D_1$
DPO-I_1	SFT-I_1	SFT-I_1	D_2
PPO-I_1	DPO-I_1	DPO-I_1	D_3
SFT-I_2	Mistral-Base	PPO-I_1	$D_0 \cup D_1 \cup D_4$
DPO-I_2	SFT-I_2	SFT-I_2	$D_2 \cup D_5$
PPO-I_2	DPO-I_2	DPO-I_2	$D_3 \cup D_6$
SFT-I_3	Mistral-Base	PPO-I_2	$D_0 \cup D_1 \cup D_4 \cup D_7$
DPO-I_3	SFT-I_3	SFT-I_3	$D_2 \cup D_5 \cup D_8$
PPO-I_3	DPO-I_3	DPO-I_3	$D_3 \cup D_6 \cup D_9$

2.4 Evaluate LLMs with WizardArena

To accurately evaluate the performance of chatbot models and predict their Elo rankings, *Arena Learning* relies on a carefully curated offline test set, which is designed to strike a balance between diversity and complexity [14, 24, 25], ensuring a comprehensive assessment of the models' capabilities across a wide range of conversational scenarios. Inspired by WizardLM [11] In-Breadth Evolving and In-Depth Evolving, we construct 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 (i.e., gte-large [26]). 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.

With the above "judge" model and the offline WizardArena test set in place, we 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 Experiments

3.1 Experimental Setup

Training Data. We random sample 10k ShareGPT data to train a initial model WizardLM-β-I_0. We then collected some instructions from open available datasets [10, 11, 17, 27, 28], and optimized them using the following steps: first, we filtered out all illegal and toxic 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 [29] for data deduplication; subsequently, we used an embedding model gte-large [26] to exclude instructions from the top 5 matches in semantic similarity with benchmarks (i.e., WizardArena, Arena-Hard Auto [24], MT-Bench [14], AlpacaEval [25], OpenLLM Leaderboard [30–34]) to prevent test data leakage. Finally, we removed all non-English instructions. After completing these steps, we obtain the refined 276K dataset D, and randomly split it to 9 parts.

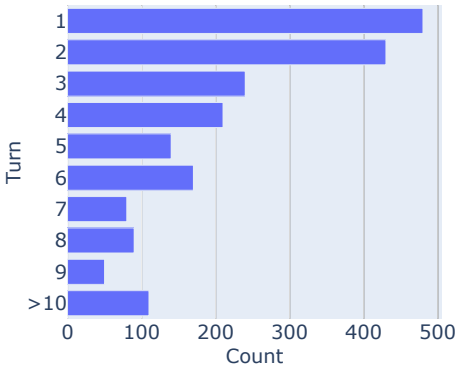

Figure 4: WizardArena-Mix Turn statistics

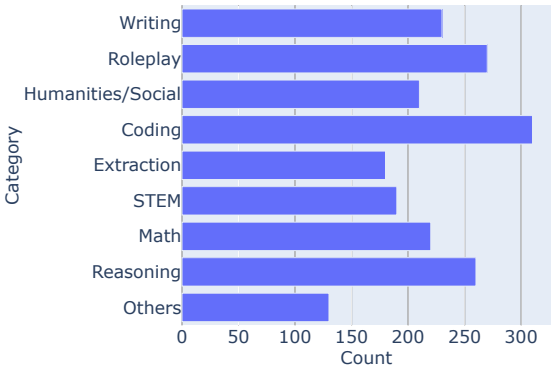

Figure 5: WizardArena-Mix Category statistics

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. The Offline-Mix WizardArena combines the Diverse and Hard test sets in 2,000 samples. Different from Arena-Hard-v1.0 [24], which mainly focuses on single-turn dialogue data, WizardArena-Mix incorporates multi-turn dialogue data. Figures 4 and 5 display the distribution of dialogue turn and the categories statistics within WizardArena-Mix, respectively. The data indicates that our multi turn conversation data accounts for a large proportion, and the distribution of topics is also diverse.

LLM Battle. We selected some popular models and conducted pairwise battles in the Offline-Mix WizardArena. Llama3-70B-Instruct [22] served as the "judge" model, with the higher-scoring model declared the winner. Following LMSYS Chatbot Arena, we adopt the Bradley-Terry model [35] to calculate the final ELO 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 [23]. 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. Table 2 contrasts the differences between WizardArena and LMSYS Arena. WizardArena leverages LLM to conduct Battles, whereas LMSYS ChatBot Arena relies on human annotation. At the same battle count, if we use sixteen 80G GPUs for inference and judgement, the process will be completed in 9 days, achieving a 40x speedup increase compared to the 12 months required by LMSYS ChatBot Arena.

Table 2: Efficiency Comparison of LMSYS Chat-Bot Arena and WizardArena.

Metrics	LMSYS ChatBot Arena	Ours
Battle Method	Human	LLM
Battle Count	1M	1M
GPU	-	16 H100
Inference Time	-	3 Days
Judge Time	~1 year	6 Days
Speed Up	1x	40x

Implementation Details. We apply our method to the Mistral-7B [36] and Mixtral-8x22B for post-training, using Llama3-70B-Instruct [22] as judge models in WizardArena. For WizardLM-β-7B, the battle models are {Command R+ [37], Qwen1.5-72B-chat [7], OpenChat-3.5 [12]}, for WizardLM-β-8x22B, the battle models are {GPT-4o [4], GPT-4-1106-preview [4], WizardLM-2-8x22B-0415 [11]}. In supervised fine-tuning, we trained three epochs 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.

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

Figure 6 and Table 4 present the rankings for some popular models across several evaluation benchmarks: LMSYS ChatBot Arena-EN [19], MT-Bench [14], and 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. In addition, there is a significant difference in performance between WizardArena diverse and hard subsets: Vicuna-33B [9] and Qwen1.5-32B-Chat [7] are more effective in diverse tasks, while Tulu-2-DPO-70B [38] and Nous-Hermes-2-Mixt-DPO [39] achieves better results in hard tasks. We therefore use WizardArena-Mix as the final evaluation benchmark of *Arena Learning* to balance the strengths of different models.

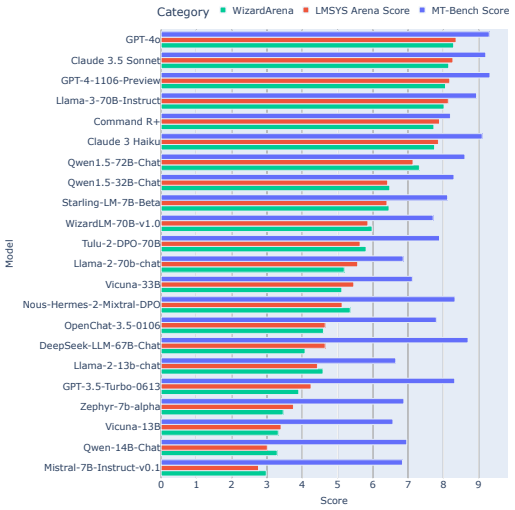

Figure 6: The performance of LLMs across MT-Bench, normalized LMSYS ChatBot Arena, and WizardArena.

Table 3: The consistency of MT-Bench, Arena-Hard-v1.0, and WizardArena compared with LMSYS ChatBot Arena. Llama-3-70B-Chat is the "Judge" model.

Metrics	MT-Bench	Arena-Hard-v0.1	WizardArena-Diverse	WizardArena-Hard	WizardArena-Mix
Data Size	160	500	1000	1000	2000
Spearman Correlation	79.36%	90.44%	98.79%	98.84%	**99.23%**
Human Agreement with 95% CI	26.04%	80.86%	97.33%	98.22%	**99.11%**
Differentiation with 95% CI	23.45%	92.33%	97.63%	96.84%	**98.02%**
Avg.	63.56%	87.88%	97.92%	97.97%	**98.79%**

Table 3 illustrates that the Offline WizardArena-Mix significantly outperforms MT-Bench across several consistent metrics which refer to the Appendix A for details: a 19.87% higher Spearman Correlation, a 73.07% increase in Human Agreement with 95% CI, and a 74.57% improvement in Differentiation with 95% CI. It achieves an average consistency of 98.79% with the LMSYS ChatBot Arena by human judgment, outperforming Arena-Hard-v1.0 [24] by 10.91% and MT-Bench [14] by 35.23%. In contrast to MT-Bench and Arena-Hard-v1.0 which use proprietary models (i.e. GPT-4) as the judge model, our approach employs current SOTA open-source model Llama-3-70B-Chat, which not only has a significantly lower cost but also achieves strong consistency. Moreover, the Offline WizardArena-Mix, which integrates both Diverse and Hard test sets, achieves 0.87% higher average consistency compared to WizardArena-Diverse and 0.82% higher than WizardArena-Hard. This indicates that balancing diversity and complexity is crucial for the effective offline evaluation of large language models. Above results also further prove the feasibility of using the "judge" model to judge the battles between LLMs and generate a large amount of post-training data in simulated arena.

Table 4: The ELO rankings on LMSYS ChatBot Arena EN (June, 2024), MT-Bench, and WizardArena. Llama-3-70B-Chat is the "judge". Llama-2-70B-Chat Elo is the reference.

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
GPT-4o [4]	1266 (+4/-4)	1401 (+3/-4)	1392 (+4/-5)	1395 (+5/-4)	9.30
Claude 3.5 Sonnet [5]	1246 (+4/-7)	1389 (+5/-6)	1378 (+6/-6)	1384 (+6/-4)	9.20
Gemini 1.5 Pro [6]	1235 (+5/-4)	1383 (+6/-5)	1373 (+5/-5)	1377 (+5/-5)	-
GPT-4-1106-Preview [4]	1232 (+3/-4)	1369 (+3/-5)	1376 (+6/-4)	1374 (+4/-3)	9.32
WizardLM-2-8x22B-0415 [11]	-	1365 (+6/-7)	1359 (+5/-7)	1361 (+5/-6)	9.12
Llama-3-70B-Instruct [22]	1227 (+3/-3)	1366 (+5/-5)	1354 (+6/-5)	1357 (+6/-4)	8.94
WizardLM-β-8x22B-I_3	-	1355 (+5/-7)	1346 (+6/-5)	1349 (+5/-7)	8.85
Command R+ [37]	1163 (+4/-4)	1351 (+9/-6)	1327 (+8/-6)	1337 (+6/-4)	8.20
Claude 3 Haiku [5]	1158 (+4/-3)	1340 (+4/-5)	1345 (+5/-5)	1342 (+4/-6)	9.10
WizardLM-β-8x22B-I_2	-	1339 (+6/-6)	1326 (+6/-8)	1332 (+6/-7)	8.49
Qwen1.5-72B-Chat [7]	1135 (+3/-4)	1332 (+9/-7)	1312 (+7/-5)	1321 (+6/-5)	8.61
WizardLM-β-8x22B-I_1	-	1325 (+8/-6)	1311 (+7/-7)	1318 (+8/-7)	7.98
Qwen1.5-32B-Chat [7]	1109 (+4/-5)	1298 (+7/-8)	1276 (+5/-8)	1283 (+6/-4)	8.30
WizardLM-β-7B-I_3	-	1269 (+5/-4)	1278 (+5/-4)	1274 (+5/-6)	8.16
Starling-LM-7B-Beta [18]	1108 (+5/-5)	1275 (+6/-4)	1270 (+6/-5)	1272 (+4/-6)	8.12
WizardLM-β-7B-I_2	-	1256 (+5/-7)	1233 (+4/-7)	1246 (+6/-5)	7.98
WizardLM-β-7B-I_1	-	1228 (+4/-6)	1201 (+6/-8)	1214 (+5/-8)	7.74
WizardLM-70B-v1.0 [11]	1098 (+7/-6)	1184 (+6/-6)	1163 (+6/-5)	1169 (+5/-5)	7.71
Tulu-2-DPO-70B [38]	1091 (+8/-10)	1147 (+8/-6)	1181 (+5/-6)	1157 (+4/-6)	7.89
Llama-2-70B-Chat [22]	1097 (+5/-4)	1100 (+0/-0)	1100 (+0/-0)	1100 (+0/-0)	6.86
Vicuna-33B [9]	1086 (+6/-5)	1113 (+5/-7)	1076 (+7/-5)	1091 (+4/-5)	7.12
Nous-Hermes-2-Mixtral-DPO [39]	1078 (+9/-8)	1107 (+8/-6)	1121 (+7/-7)	1114 (+5/-4)	8.33
OpenChat-3.5 [12]	1065 (+9/-10)	1042 (+7/-5)	1050 (+8/-5)	1045 (+5/-5)	7.80
DeepSeek-LLM-67B-Chat [40]	1065 (+12/-10)	991 (+7/-7)	1008 (+5/-7)	1000 (+7/-5)	8.70
Llama-2-13B-Chat [22]	1061 (+5/-6)	1052 (+5/-6)	1041 (+7/-7)	1042 (+5/-4)	6.65
GPT-3.5-Turbo-1106 [4]	1052 (+5/-5)	955 (+6/-7)	1004 (+6/-7)	981 (+5/-5)	8.32
Zephyr-7b-alpha [41]	1040 (+17/-13)	905 (+7/-6)	967 (+6/-8)	939 (+4/-5)	6.88
Vicuna-13B [9]	1029 (+6/-5)	934 (+6/-7)	923 (+8/-5)	927 (+5/-5)	6.57
Qwen-14B-Chat [7]	1017 (+9/-10)	916 (+5/-7)	932 (+6/-8)	924 (+4/-6)	6.96
Mistral-7B-Instruct-v0.1 [36]	1009 (+7/-7)	883 (+6/-7)	904 (+6/-9)	894 (+4/-5)	6.84
WizardLM-β-8x22B-I_0	-	873 (+5/-9)	897 (+4/-8)	889 (+4/-9)	6.78
WizardLM-β-7B-I_0	-	862 (+8/-7)	884 (+6/-7)	871 (+5/-8)	6.41

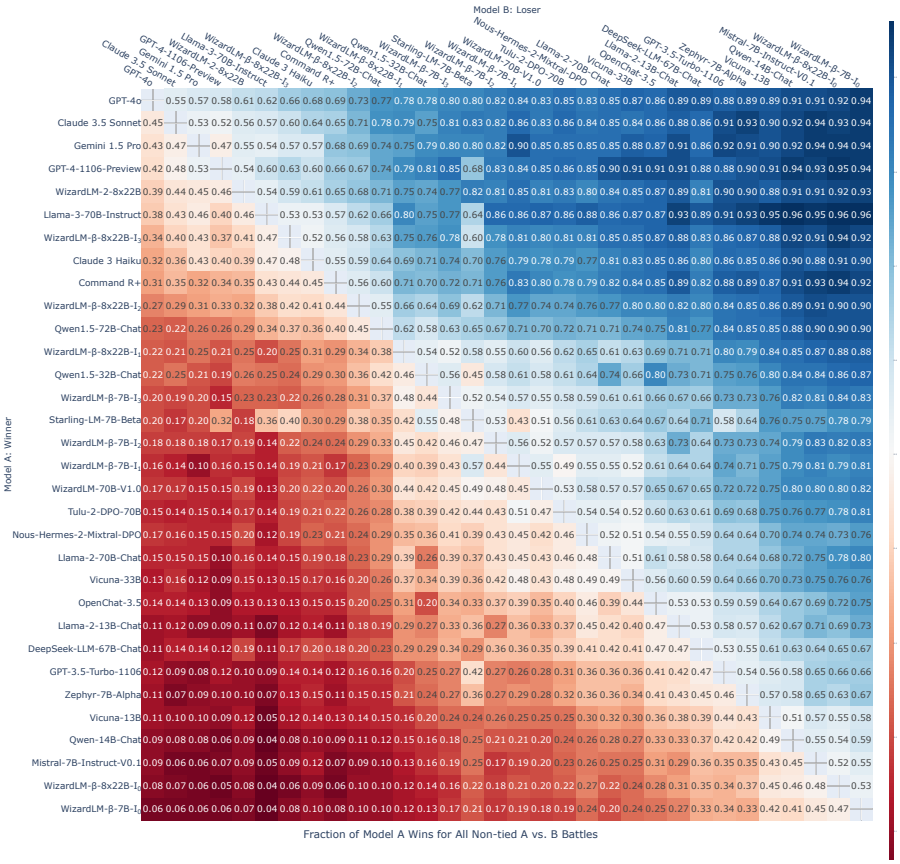

Figure 7: Win rates (w/o tie) of models in WizardArena-Mix. Each model involved in 2k x 31 battles.

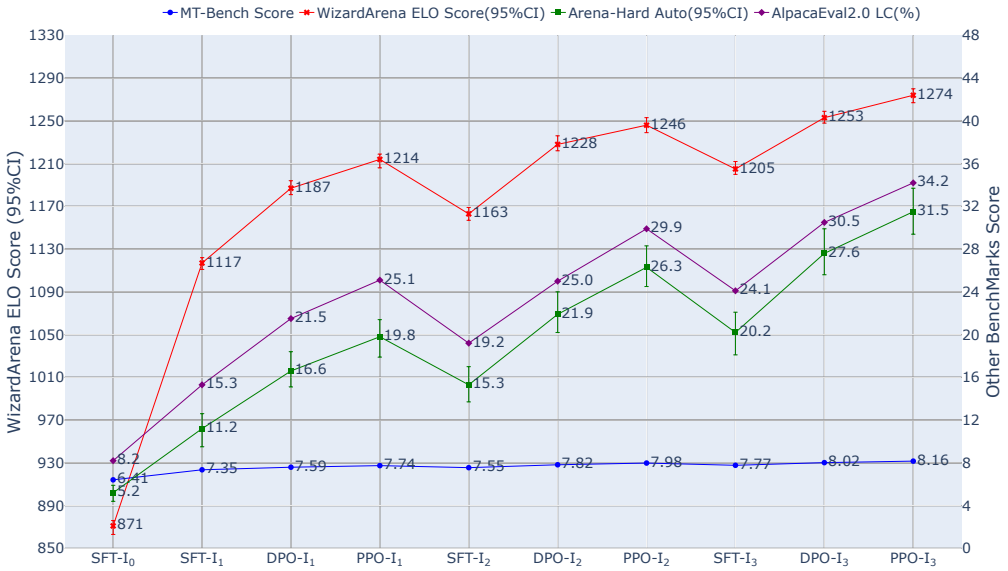

Figure 8: Explore the impact of iterative training processes of SFT, DPO, and PPO on the WizardLM-β-7B model performance in four benchmarks.

3.3 Can *Arena Learning* build an effective data flywheel with post-training?

Table 4 demonstrates the impact of using the *Arena Learning* method to post-train WizardLM-β models during three data flywheel iterations, where I_i represents the i-th iteration. In each iteration from I_1 to I_3, we always use 90k data for post-training. Starting from WizardLM-β-7B-I_0, the next 3 iterations have improved by 343 points, 32 points, and 28 points on Wizardarena-Mix Elo, respectively. At the same time, the MT-bench score of this model has also achieved significant improvement (from 6.41 to 8.16). Specifically, the WizardLM-β-7B-I_1 even surpasses WizardLM-70B-v1.0 and the WizardLM-β-7B-I_3 also shows comparable performance with Starling-LM-7B-Beta. It is worth noting that we have also observed the same trend on WizardLM-β-8x22B models, and even achieved a more significant increase in both Wizardarena-Mix Elo (+460) and MT-Bench (+2.07). This model also beats both Command R+ and Claude 3 Haiku. Figure 7 presents the win rates of 32 models in WizardArena-Mix, with each model involving in 2k x 31 battles. Compared to those baselines, our model has achieved significant improvements in win rate from the I_0 to I_3. Specifically, using GPT-4o as the battle target, our WizardLM-β-8x22B's win rate increased by 26% (8% -> 22% -> 27% ->34%), WizardLM-β-7B's win rate also increased by 14% (6% -> 16% -> 18% ->20%).

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.

3.4 Scaling Iterative SFT, DPO, and PPO with *Arena Learning* .

As the core question of this paper asks how *Arena Learning* improves a model's performance with post-training, in this section we examine how performance is affected by different post-training technology and data flywheel iterations. Figure 8 explores the results of WizardLM-β-7B model. As expected, we observe that each performance across the SFT and RL models improves step by step as we add more selected data from more *Arena Learning* battle iterations. Specifically, from SFT-I_0 to PPO-I_3, the WizardArena-Mix ELO score improves from 871 to 1274, achieves a huge gain of 403 points, and the Arena-Hard Auto ELO score also rises by 26.3 points (from 5.2 to 31.5). Additionally, the AlpacaEval 2.0 LC win rate improved by 26%, from 8.2% to 34.2%, and the MT-Bench score increased by 1.75 points, from 6.41 to 8.16. Significant improvements across four key benchmarks highlight the effectiveness and scalability of the iterative training approach proposed by *Arena Learning* in enhancing post-training LLMs during the SFT, DPO, and PPO stages.

3.5 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 during the first round of SFT stage. In Table 5, we use 10k samples for each method except for the Original D_1. The results indicate that data selected via the pair-judge method yielded a 29-point improvement in the WizardArena-Mix ELO over the all original 30k data, surpassed the diversity-based K-Means Cluster method by 23 points, and exceeded the instruction complexity-based INSTAG [43] method by 12 points. On MT-bench, the pair-judge method also demonstrated superior performance, with improvements of 0.35 points over Original Data, 0.25 points over K-Means Cluster, and 0.11 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. Simultaneously, these results underscore the effectiveness of the pair-judge method in selecting high-quality data during the SFT stage to target and strengthen the weakness of the base model.

Table 5: Explores data selection strategies during the first round of SFT stage, using 10k samples for each method except for the Original D_1.

Data Selection	Data Size	WizardArena-Mix ELO (95% CI)	MT-Bench
Original Data	30k	1079 (+5/-8)	6.88
Random Sample	10k	1072 (+8/-7)	6.77
K-Means Cluster	10k	1085 (+7/-5)	6.98
Instruction Length	10k	1081 (+5/-9)	6.92
IFD [42]	10k	1091 (+7/-6)	7.07
INSTAG [43]	10k	1096 (+5/-8)	7.12
Pair-judge	10k	1108 (+6/-8)	7.23

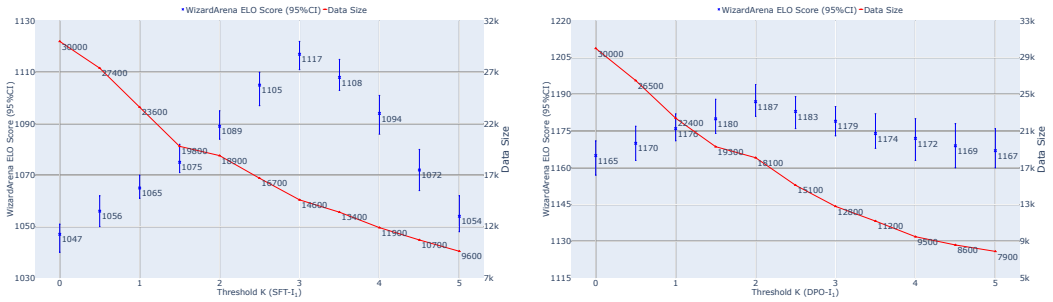

Figure 9: Explore the impact of the threshold K on the WizardLM-β-7B model during the first round of SFT and DPO.

The relationship between data size and performance. An intuitive question is whether the improvement in model performance is solely due to the increase in data size. Therefore, in this section, we discuss the impact of data size and quality on model performance. Threshold is an important hyperparameter in *Arena Learning* that controls the size of SFT data and gap between <chosen, reject> pairs of RL data. We conducted the experiments of WizardLM-β-7B-SFT-I_1 and WizardLM-β-7B-DPO-I_1 where threshold ranges from 0 to 5. The result is shown in the Figure 9, and we did observe the best threshold of SFT and DPO data are 3.0 and 2.0 respectively in I_1. In SFT, compared to threshold=0, although half of the training data (30k -> 14.6k) is left when the threshold=3, the ELO of the model actually brings a 70-point improvement (1047 -> 1117). Similarly in DPO, setting the threshold=2 reduced the data to 18.1k compared to threshold=0, and the ELO of the model improved by 22 points (1165 -> 1187). This indicates that the battle helps us filter out the truly needed data, thereby constructing a more efficient data flywheel with a more streamlined scale.

Llama3-Chat Judge or GPT-4 Judge? 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 6 explores the consistency between Llama3-70B-Instruct and GPT-4 as judge models in the WizardArena-Mix Arena. Using GPT-4 judge's ELO as the reference benchmark, the Spearman correlation coefficient between Llama3-70B-Instruct judge and GPT-4 judge is 99.26%, and the Human Agreement with 95% CI is 96.15%. The overall average consistency between the two judge models is 97.71%. Furthermore, combining GPT-4 and Llama3-70B-Instruct as the judge model resulted in an overall average consistency of 98.40% for LMSYS ChatBot Arena, a slight 0.25% improvement over using

Table 6: Explore the consistency between Llama3-70B-Instruct and GPT-4 as judging models in the Offline-Mix Arena. Using multiple bootstraps (i.e., 100), we select the median as the model's ELO score and employ Llama-2-70B-Chat ELO score as the reference point.

Model	LMSYS-ChatBot Arena-ELO-EN (95% CI)	WizardArena-Mix-ELO GPT-4-judge (95% CI)	WizardArena-Mix-ELO Llama3-70B-Instruct-judge (95% CI)	WizardArena-Mix-ELO {GPT-4 & Llama3-70B-Instruct}-judge (95% CI)
GPT-4o [4]	1266 (+4/-4)	1388 (+5/-3)	1395 (+5/-4)	1399 (+5/-4)
Calude 3.5 Sonnet [5]	1246 (+4/-7)	1372 (+6/-6)	1384 (+6/-4)	1387 (+6/-6)
Gemini 1.5 Pro [6]	1235 (+5/-4)	1365 (+4/-3)	1377 (+5/-5)	1375 (+5/-5)
Command R+ [37]	1163 (+4/-4)	1349 (+5/-7)	1337 (+6/-4)	1340 (+4/-4)
Claude 3 Haiku [5]	1158 (+4/-3)	1355 (+3/-5)	1342 (+4/-6)	1346 (+3/-4)
Qwen1.5-72B-Chat [7]	1135 (+3/-4)	1331 (+6/-5)	1321 (+6/-5)	1327 (+5/-5)
Qwen1.5-32B-Chat [7]	1109 (+4/-5)	1297 (+4/-7)	1283 (+6/-4)	1278 (+7/-4)
Starling-LM-7B-Beta [18]	1108 (+5/-5)	1275 (+6/-7)	1272 (+4/-6)	1274 (+5/-5)
WizardLM-70B-v1.0 [11]	1098 (+7/-6)	1107 (+5/-4)	1169 (+5/-5)	1166 (+6/-4)
LLama-2-70B-Chat [22]	1097 (+5/-4)	1100 (+0/-0)	1100 (+0/-0)	1100 (+0/-0)
Nous-Hermes-2-Mixtral-DPO [39]	1078 (+9/-8)	1063 (+7/-8)	1114 (+5/-4)	1109 (+7/-8)
DeepSeek-LLM-67B-Chat [40]	1065 (+12/-10)	985 (+7/-9)	1000 (+7/-5)	998 (+4/-7)
Llama-2-13B-Chat [22]	1061 (+5/-6)	974 (+7/-5)	1042 (+5/-4)	1044 (+6/-6)
GPT-3.5-Turbo-0613 [4]	1052 (+5/-5)	942 (+8/-6)	981 (+6/-5)	977 (+7/-6)
Zephyr-7b-alpha [41]	1040 (+17/-13)	925 (+5/-6)	939 (+4/-5)	937 (+4/-5)
Vicuna-13B [9]	1029 (+6/-5)	939 (+5/-8)	927 (+5/-5)	927 (+6/-6)
Qwen-14B-Chat [7]	1017 (+9/-10)	916 (+6/-6)	924 (+4/-6)	923 (+4/-6)

only Llama3-70B-Instruct(98.40% vs. 98.15%). 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.

Number of battle models. Figure 10 presents an ablation study investigating the impact of the number of other battle models. According to Table 4, the models are ranked in descending order based on WizardArena-Mix ELO scores. Subsequently, models ranging from Command R+ to OpenChat 3.5 are selected for battle. As the number of models participating in the battle increases, the performance of the WizardLM-β-7B-SFT-I_1 model gradually increases. Specifically, on WizardArena-Mix, the ELO rating of WizardLM-β-7B increases from 876 to 1159, a gain of 283 points. Concurrently, the MT-Bench score rises from 6.41 to 7.66, an increase of 1.25 points. This demonstrates the scalability of our method and its compatibility with different models, providing a basis for future large-scale application of *Arena Learning* . However, as relationship between the complexity of the battle $O(\cdot)$ and the number of models n is $O(n^2)$, and in order to balance the computational cost and model performance, we chose 3 other models to battle with WizardLM-β as the default setting in this paper.

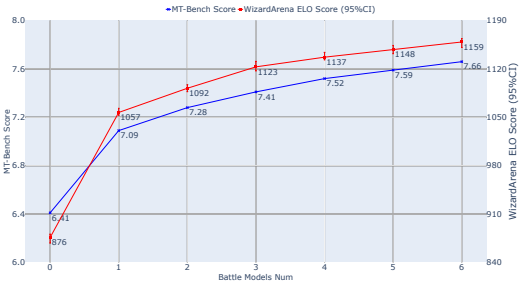

Figure 10: Explore the impact of the scale of battle models on WizardLM-β-7B-SFT-I_1.

The impact of different battle modes. In order to explore the necessity of using multiple models pairwise battle to construct a data flywheel, we designed various battle modes on D_1 SFT data, including: i) {ours + 1 other model} pairwise battle with each other, ii) randomly split D_1 into 3 parts, ours battle with one other model on each part respectively, iii) {ours + 2 other models} pairwise battle with each other, iv) {ours + 3 other models} pairwise battle with each other. We use WizardLM-β-7B-SFT-I_0, Openchat-3.5, Qwen-1.5-72B, and CommandR+ as the battle group in this section, the output model is WizardLM-β-7B-SFT-I_1. As shown in the Table 7, the mode (iv) achieved best performance on WizardArena and Outperformed the (i) mode {Only Command R+ battle} by 89 points and the (iii) mode {Command R+ & Qwen1.5-72B-Chat Battle} by 22 points. To this end, we finally leverage multiple models pairwise battle with each other to build the simulated offline Chatbot Arena.

Table 7: The WizardArena Elo of WizardLM-β-7B-SFT-I_1 on different battle modes.

Battle Mode	WizardArena
i) Ours v.s. OpenChat-3.5	924 (+7/-5)
i) Ours v.s. Qwen-1.5-72B	1015 (+5/-5)
i) Ours v.s. Command R+	1028 (+6/-4)
ii) Ours v.s. {Qwen-1.5-72B/OpenChat-3.5/Command R+}	1046 (+5/-8)
iii) {Ours, Qwen-1.5-72B, OpenChat-3.5}, 1v.s.1	1052 (+6/-7)
iii) {Ours, Command R+, OpenChat-3.5}, 1v.s.1	1065 (+5/-8)
iii) {Ours, Qwen-1.5-72B, Command R+}, 1v.s.1	1095 (+5/-5)
iv) {Ours, Qwen-1.5-72B, Command R+, OpenChat-3.5}, 1v.s.1	1117 (+5/-6)

Performance on more benchmarks. Table 8 highlights the performance of WizardLM-β across various metrics after three iterations, including LMSYS Arena-Hard Auto, AlpacaEval 2.0 LC, and the

Table 8: Explore the performance of the WizardLM-β model across various benchmarks. The results of baselines are cited from Arena-Hard Auto [24], AlpacaEval 2.0 LC [25], and OpenLLM Leaderboard [30].

Model	**Arena-Hard Auto (95% CI)**	**AlpacaEval 2.0 LC (Win Rate %)**	**ARC**	**Hellaswag**	**MMLU**	**TruthfulQA**	**Avg.**
Claude 3.5 Sonnet [5]	79.3 (-2.1, 2.0)	52.4	-	-	-	-	-
GPT-4o [4]	79.2 (-1.9, 1.7)	57.5	-	-	-	-	-
GPT-4-0125-Preview [4]	78.0 (-2.1, 2.4)	-	-	-	-	-	-
Gemini 1.5 Pro [6]	72.0 (-2.1, 2.5)	-	-	-	-	-	-
WizardLM-2-8x22B-0415 [11]	69.6 (-1.8, 2.4)	51.3	-	-	-	-	-
GLM-4-0520 [44]	63.8 (-2.9, 2.8)	-	-	-	-	-	-
Yi-Large [45]	63.7 (-2.6, 2.4)	51.9	-	-	-	-	-
DeepSeek-Coder-V2-Instruct [46]	62.3 (-2.1, 1.8)	-	-	-	-	-	-
Gemma-2-27B-it [47]	57.5 (-2.1, 2.4)	-	-	-	-	-	-
GPT-4-0314 [4]	50.0 (0.0, 0.0)	35.3	-	-	-	-	-
Qwen2-72B-Instruct [7]	46.9 (-2.5, 2.7)	-	-	-	-	-	-
Claude 3 Sonnet[5]	46.8 (-2.3, 2.7)	34.9	-	-	-	-	-
Llama-3-70B-Instruct [22]	41.1 (-2.0, 2.2)	34.4	71.42	85.69	80.06	61.81	74.75
Mixtral-8x22b-Instruct-v0.1 [36]	36.4 (-2.4, 2.6)	30.9	72.70	89.08	77.77	68.14	76.92
Qwen1.5-72B-Chat [7]	36.1 (-2.0, 2.7)	36.6	68.26	86.47	77.46	63.84	74.01
Phi-3-Medium-4k-Instruct [48]	33.4 (-2.6, 2.1)	-	67.32	85.76	77.83	57.71	72.16
Command R+ [37]	33.1 (-2.8, 2.4)	-	70.99	88.56	75.73	56.30	72.90
GPT-3.5-Turbo-0613 [4]	24.8 (-1.9, 2.3)	22.7	-	-	-	-	-
DBRX-Instruct [49]	23.9 (-1.5, 1.5)	25.4	67.83	88.85	73.72	67.02	74.36
Yi-34B-Chat [45]	23.1 (-1.6, 1.8)	27.2	70.48	85.97	77.08	62.16	73.92
Phi-3.1-Mini-4k-Instruct [48]	23.1 (-2.4, 2.0)	-	62.97	80.6	69.08	59.88	68.13
Starling-LM-7B-Beta [18]	23.0 (-1.8, 1.8)	-	67.24	83.47	65.14	55.47	67.83
Llama-3-8B-Instruct [22]	20.6 (-2.0, 1.9)	22.9	60.75	78.55	67.07	51.65	64.51
Tulu-2-DPO-70B [38]	15.0 (-1.6, 1.3)	21.2	72.10	88.99	69.84	65.78	74.18
Mistral-7B-Instruct-v0.1 [36]	12.6 (-1.7, 1.4)	-	54.52	75.63	55.38	56.28	60.45
Llama-2-70B-Chat [22]	11.6 (-1.5, 1.2)	14.7	64.59	85.88	63.91	52.80	66.80
Vicuna-33B [9]	8.6 (-1.1, 1.1)	17.6	62.12	83.00	59.22	56.16	65.13
Gemma-7B-it [47]	7.6 (-1.2, 1.3)	10.4	51.45	71.96	53.52	47.29	56.06
Llama-2-7b-chat [22]	4.6 (-0.8, 0.8)	5.4	52.90	78.55	48.32	45.57	56.34
Nous-Hermes-2-Mixtral-DPO [39]	-	-	71.42	87.21	72.28	54.53	71.36
DeepSeek-LLM-67B-Chat [40]	-	17.8	67.75	86.8	72.19	55.83	70.64
OpenChat-3.5-0106 [12]	-	-	66.04	82.93	65.04	51.90	66.48
Zephyr-7b-beta [41]	-	13.2	62.03	84.36	61.07	57.45	66.23
Qwen1.5-7B-Chat [7]	-	14.7	55.89	78.56	61.65	53.54	62.41
Vicuna-13b-v1.5 [9]	-	11.7	57.08	81.24	56.67	51.51	61.63
Llama-2-13B-Chat [22]	-	8.4	59.04	81.94	54.64	44.12	59.94
WizardLM-β-7B-I_0	5.2 (-0.8, 0.7)	8.2	54.73	72.67	54.43	49.16	57.75
WizardLM-β-7B-I_1	19.8 (-1.9, 1.6)	25.1	60.32	83.11	61.50	55.92	65.21
WizardLM-β-7B-I_2	26.3 (-1.8, 2.0)	29.9	62.25	84.38	63.96	56.67	66.82
WizardLM-β-7B-I_3	31.5 (-2.1, 2.2)	34.2	64.58	84.93	65.74	57.06	68.08
WizardLM-β-8x22B-I_3	64.3 (-2.0, 2.5)	48.9	67.91	86.64	73.76	66.48	73.70

OpenLLM Leaderboard. In LMSYS Arena-Hard Auto, WizardLM-β-7B's score rises from 5.2 to 31.5, with a gain of 26.3 points, surpassing GPT-3.5-Turbo-0613 by 6.7 points and Llama 3-8B-Instruct by 10.9 points, closely aligning with Command R+. WizardLM-β-8x22B's performance outperforms Llama-3-70B-Instruct by 23.2 points, is also better than GLM-4-0520 and Yi-Large. In AlpacaEval 2.0 LC, WizardLM-β-7B's win rate increases from 8.2% to 34.2%, exceeding GPT-3.5-Turbo-0613 by 11.5 points and Mixtral-8x22b-Instruct-v0.1 by 3.3 points, matching closely with Llama3-70B-Instruct. Moreover, WizardLM-β-8x22B's win rate even surpasses Llama-3-70B-Instruct by 14.5 points and GPT-4-0314 by 13.6 points. On the OpenLLM Leaderboard, WizardLM-β-7B's average score increases from 57.75 to 68.08, surpassing Llama-2-70B-Chat by 1.28 points and comparable to Starling-LM-7B-beta. WizardLM-β-8x22B is also compareable with Command R+, exceeds Deepseek-LLM-67B-Chat by 3.06 points, and closely approaches Qwen1.5-72B-Chat and Llama-3-70B-Instruct. The above results indicate that: 1) Utilizing the *Arena Learning* method to generate training data significantly improves the performance of the model by multiple training iterations. 2) *Arena Learning* can improves the generalization and scalability of the model performance.

Table 9: Data count and difficulty of each iteration.

	Threshold	Count	Difficulty
Original	-	30k x 3	4.7
SFT-I_1	3.0	14.6k	5.8
SFT-I_2	1.0	11.3k	6.5
SFT-I_3	1.0	7.8k	7.4
SFT-Total	-	33.7k	6.4

Data count and difficulty of each iteration. In table 9 we show in detail the data size, difficulty, and threshold division for each round of the SFT. As the number of iteration rounds increased, we adjusted the threshold from 3 to 1, but the data size of SFT still significantly decreased (30k -> 7.8k). This is because as the model's ability evolved, the number of battles it lost also sharply declined. We also found that the difficulty of each round of data gradually increases (4.7 -> 7.4) and we only need totally around 1/3 data for final SFT (90k -> 33.7k) and the average difficulty

is 6.4. It indicates that a reasonable data flywheel should focus more on finding those challenging data for target model to fill in the shortcomings of its capabilities.

Table 10: Explore the quantity of selected responses for each battle model across various rounds during the SFT and DPO stages.

Stage	Command R+	Qwen1.5-72B-Chat	OpenChat-3.5	WizardLM-β-7B	Total
SFT-I_1	6.9k	5.5k	2.2k	-	14.6k
SFT-I_2	5.8k	4.2k	1.3k	-	11.3k
SFT-I_3	4.1k	3.0k	0.7k	-	7.8k
SFT-$Total$	16.8k	12.7k	4.2k	-	33.7k
DPO-I_1	8.7k	7.6k	1.9k	1.1k	19.3k
DPO-I_2	8.0k	7.2k	1.1k	1.6k	17.9k
DPO-I_3	7.4k	6.5k	0.6k	2.3k	16.8k
DPO-$Total$	24.1k	21.3k	3.6k	5.0k	54.0k

Count of data selected from each battle model. Table 10 illustrates the count of selected win/accepted responses from each battle model across 3 rounds within the SFT and DPO stages. During the SFT stages, data volume consistently declines through successive iteration rounds (14.6k -> 7.8k). Moreover, the volume of selected data strong correlates with battle model performance. For instance, Command R+ consistently requires more data than both Qwen1.5-72B-Chat and OpenChat-3.5 (16.8k > 12.7k > 4.2k). During DPO, most other battle models always show a decreasing trend in selected data per iteration round, except for WizardLM-β, which experienced an increase in data volume (1.1k -> 1.6k -> 2.3k), this is mainly because as our model performance improves, the proportion of its recovery in positive samples also increases gradually.

Data category count of each iteration. Figure 11 illustrates the selected training data size trend for SFT across various categories during each iteration. As iterations progress, there is a consistent decline in selection across all categories. However, this decline occurs more gradually in complex categories (i.e., Mathematics, Reasoning, and Coding) while it is more pronounced in simpler categories like Writing and Extraction. Specifically, by the third iteration, the proportion of selections from more challenging categories like Coding, Math, and Reasoning has increased, whereas it has decreased for less demanding categories such as Writing and Roleplay. This pattern suggests that the selection of data progressively favors more complex tasks with each iteration, thereby significantly improving the model's performance in these intricate categories.

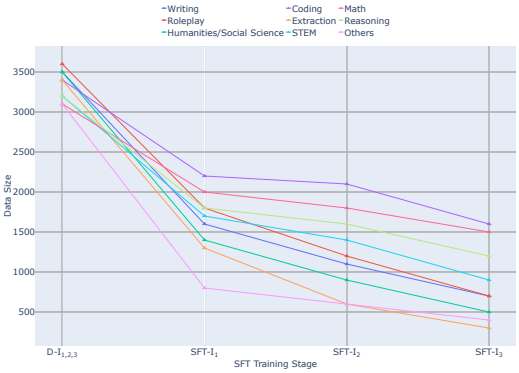

Figure 11: The selected training data size trend for SFT across each category during each iteration.

Model performance changes of each category. Figure 12 illustrates the evolution of ELO scores for the WizardLM-β-7B model across eight categories with increasing iterations during the training stage. Initially, the ELO score of WizardLM-β-7B is inferior to OpenChat 3.5. After multiple iterations, WizardLM-β-7B not only surpasses OpenChat-3.5 but also consistently approaches the performance of Qwen1.5-72B-Chat and Command R+. From iterations I_0 to I_3, the ELO scores of the model improve sharply across all categories, followed by a steady growth, indicating its gradual evolution from a weaker model to a stronger model. Particularly, in less challenging categories (i.e., Roleplay and Extraction), WizardLM-β-7B begins behind but eventually outperforms Qwen1.5-72B-Chat. Conversely, in more complex reasoning tasks like Math and Coding, its progress is slower. Moreover, the ELO battle results highlight the distinct strengths of each model. For instance, Command R+ excels in the challenging categories like Coding and Math. Meanwhile, Qwen1.5-72B-Chat shows stronger performance in Humanities/Social Science and STEM, while OpenChat3.5 is comparatively weaker. As iterations increase, training data shifts towards more complex data (i.e., Coding and Math), enhancing the model initial weaknesses. Over three rounds of iterations, our model can scale up with an extensive amount of battle training data from WizardArena, leading to substantial performance improvements. These findings highlight the significant advantages and potential of *Arena Learning* to boost post-training performance of WizardLM-β-7B by harnessing the collective knowledge and capabilities of multiple advanced models.

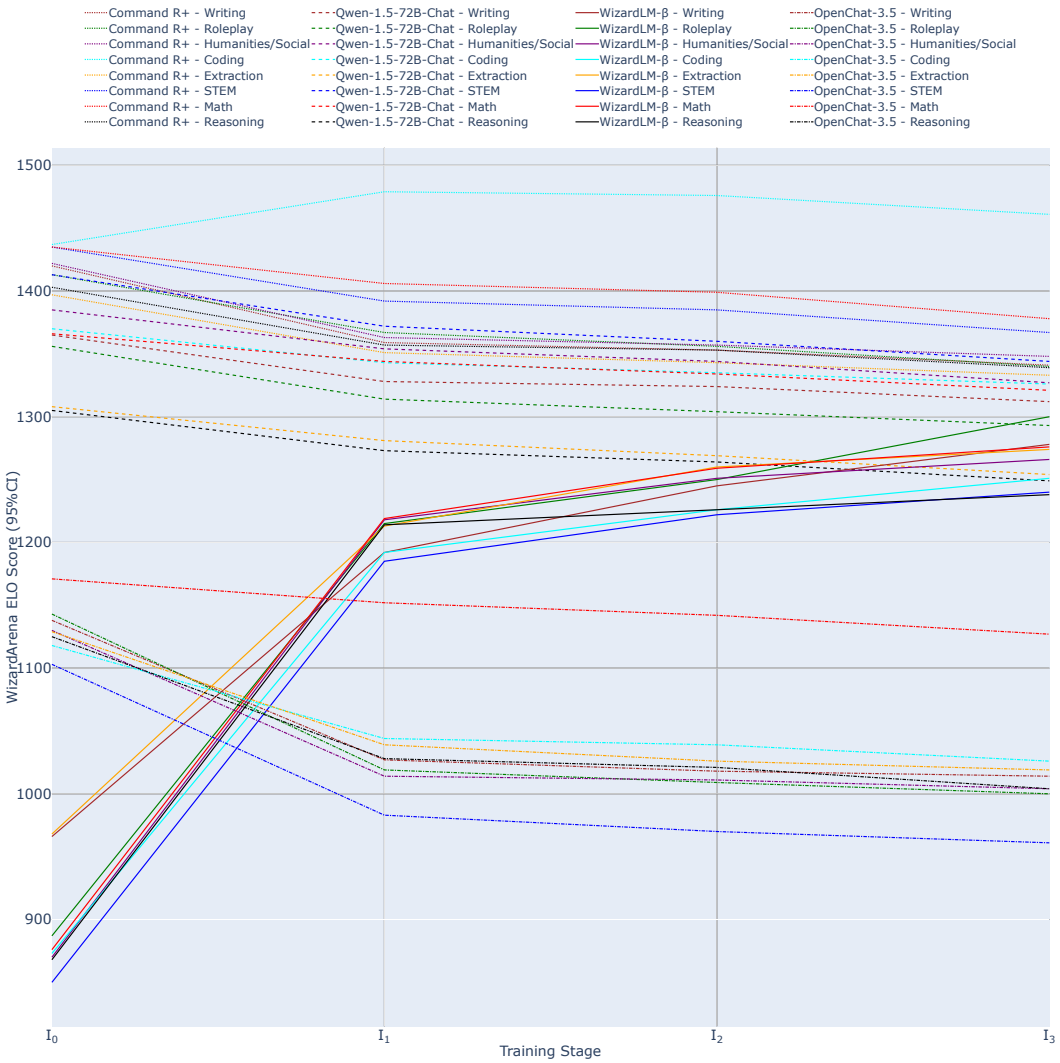

Figure 12: Explore the progression of ELO scores for the WizardLM-β-7B model across eight categories as iterations increase.

Table 11: Explore the performance impact of employing more advanced models to battle with WizardLM-β-7B-I_0 on different stages.

Training Stage	WizardArena Elo	MT-Bench
SFT-I_0	871 (+5/-8)	6.41
Battles With M_0={Command R+, Qwen1.5-72B-Chat, and OpenChat 3.5}		
SFT-I_1	1117 (+5/-6)	7.35
{SFT + DPO}-I_1	1187 (+7/-6)	7.59
{SFT + DPO + PPO}-I_1	1214 (+5/-8)	7.74
Battles With M_1={GPT-4o, GPT4-1106-Preview, and WizardLM-2-8x22B}		
SFT-I_1	1164 (+4/-7)	7.60
{SFT + DPO}-I_1	1232 (+6/-6)	7.78
{SFT + DPO + PPO}-I_1	1266 (+6/-4)	7.89

Learning from more advanced models. Table 11 analyzes the performance impact of employing more advanced models to battle for WizardLM-β-7B. Initially, leveraging the M_1 models ={GPT-4o, GPT-4 Turbo, and WizardLM-2-8x22B} in the first round improve the ELO score from the baseline SFT-I_0 of 871 to 1266, a gain of 395 points and represent a 52-point improvement over batting with the M_0 models={Command R+, Qwen1.5-72B-Chat, and OpenChat 3.5} . Throughout various stages of the battle and training, the ELO scores using the M_1 models are always correspondingly 45 ~55 points higher than the M_0 models. Additionally, the MT-Bench score increased from 6.41 to 7.89,

marking a 0.15 point advance over M_0 models score of 7.74. The results highlight the substantial performance improvements that can be achieved by employing more advanced models for battle.

4 Related Works

4.1 Large Language Models

LLMs have made significant strides in Natural Language Processing (NLP), serving as a versatile foundation for numerous applications [50–52]. 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, 53], Anthropic's Claude [54], Google's PaLM [55, 56], Gemini [6], Gemma [47], and DeepMind's Chinchilla [57]. 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 [58], Mistral AI's Mistral [36], Microsoft's Phi [48], Meta's Llama family [3, 22, 59] and GAL [60], NVIDIA's Nemotron-4 340B [61], Tsinghua University's ChatGLM [62, 63], and TII's Falcon [64] . New entries such as Command R [37], DBRX [49], Reka [65], Baichuan [66], Qwen [7], Yi [45], DeepSeek [40], InternLM [67], MiniCPM [68] and Llemma [69] have also emerged. Presently, models like Alpaca [10], Vicuna [9], Guanaco [70], Orca [71], OpenChat [12], Tulu2 [38], WizardLM [11], XwinLM [72, 73], StarlingLM [18] and Zephyr [41] are being developed through supervised fine-tuning based on Llama [3, 22, 59] and Mistral [36]. However, how to measure the performance of current all models in real-world, open scenarios is a challenging task. LMSYS has developed a chatbot arena [19] that utilizes anonymous battle and human judgment, but assessing all models is both time-consuming and costly. In this paper we simulate an offline chatbot arena and employ advanced LLM (i.e., Llama3-70B-Chat [59]) for judgment, significantly improving efficiency and reducing time requirements by 40x.

4.2 LLM Post-training

The alignment performance of Large Language Models (LLMs) is significantly influenced by the quality of Supervised Fine-Tuning (SFT) data, which encompasses task difficulty [71], query complexity [11, 74, 75], semantic diversity [10, 13], and sample size [76]. For instance, [10] generates diverse queries through self-instruct [77] methods, while [11, 74, 75, 78] enhances model alignment by increasing query complexity. [71] boosts NLP task performance by optimizing FLAN [27] queries and responses with specialized LLMs, and [13] has introduced UltraChat. To select data efficiently, some strategies like IFD [42], INSTAG [43], DEITA [79], MODS [80], and ALPAGASUS [81] are adopted. [71] employs ChatGPT to label instructional data, ensuring both diversity and complexity. Here, we select training data using the "judge pair" method with different advanced models.

To better adapt to preferences beyond SFT, models are trained with feedback-based methods like RLHF and RLAIF [2, 22, 54, 82, 83], employing Proximal Policy Optimization (PPO) [84] to align with model preferences. [85–87] improve weak to strong model generalization. WizardMath [75] adopts RLEIF, introducing process supervision and instruction quality scoring reward model to improve the mathematical reasoning ability of large language models. Due to RLHF's complexity and instability, simpler alternatives like DPO [20], RRHF [88], KTO [89], IPO [90], sDPO [91], and ORPO [92] are utilized. DPO [20] merges reward modeling with preference learning. RRHF [88] uses ranking loss to prioritize preferred answers, and KTO [89] operates without needing paired preference datasets. In this paper, in order to efficiently manage massive data, we have established a dynamic data flywheel for model post-training through the pair-wise judge battle method to consistently collect feedback from the advanced models. Furthermore, we propose *Arena Learning* to perform iterative battle and training process (SFT-DPO-PPO), where the WizardLM-β is continuously updated and re-evaluated against the SOTA models, progressively enhancing the performance of our model.

4.3 LLM Benchmarks

Large Language Models (LLMs) have transformed the way people interact with computing systems and are extensively used in everyday life and work [50]. The existing benchmarks [93–95] are mainly divided into two categories: 1) Specialized tasks. Knowledge and Capability: MMLU [32], CMMLU [96], and C-Eval [97]; Reasoning: ARC [98], HellaSwag [33], PIQA [99], GSM8k [100], MATH [101]; Programming: HumanEval [102], MBPP [103], LiveCodeBench [104]; Safety and

Truthfulness: ToxicChat [105], OLID [106], BIG-Bench [107], TruthfulQA [34]. They focus on assessing LLM performance in specific areas. 2) General tasks: like MT-Bench [14, 108] and AlpacaEval [25, 109, 110], 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 [19] and Allenai WildBench [111]) use anonymous battles, ELO [16, 112] 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. [113, 114] propose an automatic evaluation tool for instruction-tuned LLMs. Additionally, most models overfit on leaderboards like MT-Bench [14], OpenLLM leaderboard [30, 115], showing inconsistent performance with real-world ChatBot scenarios and low differentiation among models. Therefore, we have developed the simulated offline WizardArena, which not only effectively differentiates model performance but also aligns closely with the online human-based LMSYS ChatBot Arena [19], which achieves an average consistency of 98% with LMSYS ChatBot Arena, simultaneously making it suitable for selecting the optimal models and predicting the performance of models while significantly enhancing model post-training through battle data.

5 Conclusion

This paper introduces *Arena Learning*, a simulated offline chatbot arena that utilizes AI LLMs to bypass the manual and time-intensive cost typically associated with preparing the arena battle data, while preserving the core advantages of the arena-based evaluation and training. The effectiveness of *Arena Learning* is validated through the high consistency in predicting Elo rankings across various LLMs compared, when compared with the human-based LMSys Chatbot Arena. Furthermore, the model trained iteratively on synthetic data generated by *Arena Learning* exhibits significant performance improvements using various training strategies. Overall, *Arena Learning* emerges as a cost-effective and reliable alternative to conventional human-based evaluation systems, providing a sustainable approach to progressively enhance and scale the capabilities of large language 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 generate potentially unethical or misleading information.

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] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *ArXiv*, abs/2308.03281, 2023.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] Cohere Inc. Cohere: Large language models for your business, 2024.

[38] 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.

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

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

[45] 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.

[46] Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y Wu, Yukun Li, Huazuo Gao, Shirong Ma, et al. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. *arXiv preprint arXiv:2406.11931*, 2024.

[47] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[48] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[49] Mosaic Research Team et al. Introducing dbrx: A new state-of-the-art open llm, 2024. *URL https://www. databricks. com/blog/introducing-dbrx-new-state-art-open-llm. Accessed on April*, 26, 2024.

[50] 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.

[51] 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.

[52] 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.

[53] 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 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] Meta AI. Introducing meta llama 3: The most capable openly available llm to date, 2024.

[60] 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.

[61] Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*, 2024.

[62] 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.

[63] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*, 2021.

[64] 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.

[65] Aitor Ormazabal, Che Zheng, Cyprien de Masson d'Autume, Dani Yogatama, Deyu Fu, Donovan Ong, Eric Chen, Eugenie Lamprecht, Hai Pham, Isaac Ong, et al. Reka core, flash, and edge: A series of powerful multimodal language models. *arXiv preprint arXiv:2404.12387*, 2024.

[66] Baichuan. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023.

[67] InternLM Team. Internlm: A multilingual language model with progressively enhanced capabilities. `https://github.com/InternLM/InternLM`, 2023.

[68] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

[69] 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.

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

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

[72] Bolin Ni, JingCheng Hu, Yixuan Wei, Houwen Peng, Zheng Zhang, Gaofeng Meng, and Han Hu. Xwin-lm: Strong and scalable alignment practice for llms. *arXiv preprint arXiv:2405.20335*, 2024.

[73] Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. Common 7b language models already possess strong math capabilities. *arXiv preprint arXiv:2403.04706*, 2024.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] Weihao Zeng, Can Xu, Yingxiu Zhao, Jian-Guang Lou, and Weizhu Chen. Automatic instruction evolving for large language models. *arXiv preprint arXiv:2406.00770*, 2024.

[79] 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.

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

[81] 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.

[82] 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.

[83] 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.

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

[85] Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

[86] Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, and Yaodong Yang. Aligner: Achieving efficient alignment through weak-to-strong correction. *arXiv preprint arXiv:2402.02416*, 2024.

[87] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

[88] 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.

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

[90] 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.

[91] 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.

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

[93] 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.

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] 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.

[102] 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.

[103] 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.

[104] 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.

[105] 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.

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

[107] 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.

[108] 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.

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

[110] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2023.

[111] 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.

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

[113] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024.

[114] Ruochen Zhao, Wenxuan Zhang, Yew Ken Chia, Deli Zhao, and Lidong Bing. Auto arena of llms: Automating llm evaluations with agent peer-battles and committee discussions. *arXiv preprint arXiv:2405.20267*, 2024.

[115] Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*, 10:8–9, 2021.

A 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 employ several key criteria [24] 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.

$$CI_1 \cap CI_2 = \emptyset$$