WILDBENCH: Benchmarking LLMs with Challenging Tasks from Real Users in the Wild

Bill Yuchen Lin[♡] **Yuntian Deng**[♡] **Khyathi Chandu**[♡] **Faeze Brahman**[♡]
Abhilasha Ravichander[♡] **Valentina Pyatkin**[♡] **Nouha Dziri**[♡]
Ronan Le Bras[♡] **Yejin Choi**[♡♢]

[♡]Allen Institute for AI [♢]University of Washington

https://hf.co/spaces/allenai/WildBench

Abstract

We introduce WildBench, an automated evaluation framework designed to benchmark large language models (LLMs) using challenging, real-world user queries. WILDBENCH consists of 1,024 tasks carefully selected from over one million human-chatbot conversation logs. For automated evaluation with WILDBENCH, we have developed two metrics, WB-Reward and WB-Score, which are computable using advanced LLMs such as GPT-4-turbo. WILDBENCH evaluation uses task-specific checklists to evaluate model outputs systematically and provides structured explanations that justify the scores and comparisons, resulting in more reliable and interpretable automatic judgments. WB-Reward employs fine-grained pairwise comparisons between model responses, generating five potential outcomes: much better, slightly better, slightly worse, much worse, or a tie. Unlike previous evaluations that employed a single baseline model, we selected three baseline models at varying performance levels to ensure a comprehensive pairwise evaluation. Additionally, we propose a simple method to mitigate length bias, by converting outcomes of "slightly better/worse" to "tie" if the winner response exceeds the loser one by more than K characters. WB-Score evaluates the quality of model outputs individually, making it a fast and cost-efficient evaluation metric. WILDBENCH results demonstrate a strong correlation with the human-voted Elo ratings from Chatbot Arena on hard tasks. Specifically, WB-Reward achieves a Pearson correlation of 0.98 with top-ranking models. Additionally, WB-Score reaches 0.95, surpassing both ArenaHard's 0.91 and AlpacaEval2.0's 0.89 for length-controlled win rates, as well as the 0.87 for regular win rates.

1 Introduction

Large language models (LLMs) have become integral to a wide range of real-world applications due to their strong generalization capabilities across diverse tasks. However, effectively evaluating their performance remains a challenging problem, particularly when striving for an automated and cost-effective solution. Traditional benchmarking datasets like MMLU [13] focus primarily on assessing the reasoning abilities of LLMs using multiple-choice questions, which fall short in evaluating the more open-ended problems that real-world users pose. Chatbot Arena [4] provides an online platform where human preferences are collected to judge pairs of model outputs, subsequently ranking LLMs using Elo ratings. While this human-based evaluation method offers valuable insights into user preferences, it has notable limitations, such as high labor costs, the inability to deliver real-time results, a lack of data transparency, and the challenge of fairly evaluating all models with same data.

Several automated benchmarks such as AlpacaEval [15], MT-bench [38], and ArenaHard [14] employ advanced LLMs like GPT-4-Turbo to assess the quality of model responses. Comparative analyses of these benchmarks are presented in Table 1 and Figure 3. These existing benchmarks exhibit

Preprint. Under review.

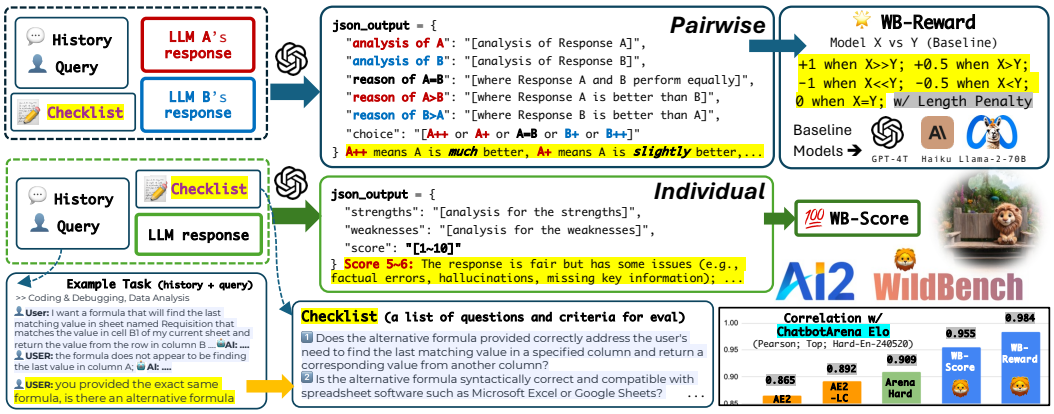


Figure 1: Evaluation framework for WILDBENCH.

significant shortcomings in task composition and skill coverage, particularly in mirroring the natural distribution of real-world user tasks. MT-bench, comprising only 80 hand-crafted examples, lacks sufficient breadth for a comprehensive evaluation. Meanwhile, AlpacaEval, with 805 tasks derived from multiple alignment datasets, includes relatively simple tasks, such as "*What is the capital of Australia?*" and suffers from low task diversity; for instance, over 20 tasks redundantly assess recipe generation skills. This benchmark mostly focuses on information-seeking tasks, containing merely 6% coding and 3% mathematics tasks. Conversely, ArenaHard, sampling 500 tasks from ChatbotArena, displays an excessive concentration on coding and debugging tasks, accounting for over 57% of its content. Most existing benchmarks do not sufficiently challenge the models with varied and unexpected nature of user inquiries in practical settings, thus limiting their overall effectiveness in providing a holistic evaluation. This issue highlights the necessity for more comprehensive benchmarks that can better simulate the wide range of tasks from real users.

In this paper, we introduce WILDBENCH, an automated evaluation framework engineered to assess LLMs using complex queries from real-world users. The examples in WILDBENCH are periodically updated, with the current version (V2) comprising 1,024 tasks carefully curated from real user-chatbot dialogs provided by the WildChat project [37]. We engage multiple advanced LLMs to process a filtered selection from WildChat, tasking them with the analysis of the requisite knowledge and skills for each task and subsequently labeling the difficulty level. Tasks considered as easy by all models are excluded. We ensure the distribution of tasks mirrors the original WildChat data, such that the task distribution of WildBench is still natural (Figure 3). Additionally, all finalized tasks undergo manual review. Further details are provided in Section 2.

WILDBENCH evaluation is illustrated in Figure 1. To design a reliable automatic evaluation, we employ two key designs for prompting LLMs as judges. Drawing inspiration from how humans evaluate responses to open-ended questions, we develop task-specific checklists. These checklists guide LLMs in generating consistent and reliable judgments, with each checklist comprising questions focused on specific criteria. Similar to the zero-shot Chain-of-Thoughts (CoT) prompting [12], we prompt LLMs to provide step-by-step, structured analyses of each LLM response. This method encourages a detailed, fine-grained evaluation process, culminating in a well-justified final decision.

To report performance, we employ two primary metrics: WB-Reward for pairwise comparisons and WB-Score for individual scoring. WB-Reward is based on pairwise comparisons between LLMs, with five possible outcomes: "A is much/slightly better/worse than B" or "Tie." Notably, we used three baseline models to compare with each testing model instead of using a single baseline model, as most prior works do. This approach provides a more comprehensive assessment based on different levels of model performance. WB-Score measures the quality of each model's generation individually, offering a quicker and more cost-effective evaluation. To mitigate the bias towards longer outputs, a common issue in LLM-as-a-judge evaluations [6], we introduced a simple length-penalty method, converting slight wins/losses to ties when the winner's output is significantly longer than the loser's.

Both metrics have demonstrated strong correlations with human judgments, evidenced by a Pearson correlation of 0.98 for WB-Reward and 0.95 for WB-Score against the human-voted Elo rating from Chatbot Arena on the top-ranking models. These scores significantly surpass other benchmarks, such

Table 1: Statistical comparison of LLM alignment benchmarks.

Dataset	#Tasks	#Turns	ChatHistory	QueryLen	PromptLen	RealUser	TaskTag	Evaluation
MT-Bench	80	2	✔Dynamic	202.2	Dynamic	✖	✔	Score
AlpacaEval	805	1	✖	164.9	164.9	✖	✖	Pair (1ref)
ArenaHard	500	1	✖	406.4	406.4	✔	✖	Pair (1ref)
WILDBENCH	1,024	≤5	✔Static	978.5	3402.1	✔✔	✔	Score+Pair (3refs)

as ArenaHard[14]'s 0.91 and AlpacaEval2.0's 0.87 (0.89 for LC)[15, 6], validating WILDBENCH's effectiveness and alignment with human-based evaluation. More details are shown in Tab 3 in Sec 4.

2 WILDBENCH Data Curation

In this section, we describe the data curation process for the tasks used to evaluate LLMs in WILDBENCH . Our goal is to ensure that the selected tasks not only represent real-world use cases but are also challenging enough to distinguish the varying capabilities of LLMs.

2.1 Mining Challenging Tasks from WildChat

We sourced tasks from the WildChat dataset [37], which comprises one million human-chatbot conversations from real users.[1] This dataset is particularly suited for conversion into an evaluation benchmark because it contains a diverse array of tasks that users expect LLMs to perform, such as writing assistance, coding, mathematics, data analysis, role playing, and planning.

Basic filtering. To control the quality and diversity of the selected tasks, we applied several filtering steps. First, we removed user queries that were either too short (less than 10 tokens) or excessively long (more than 3,000 tokens). We also excluded conversations with more than five user-chatbot turns to maintain focus and coherence in the tasks, as conversations exceeding five turns tend to contain multiple topics. Furthermore, we focused on English data and filtered out non-English tasks. Since our focus is more on evaluating the capabilities of LLMs rather than content moderation, we also removed toxic conversations. To ensure task diversity, we used sentence embeddings from SentenceBERT [30] to calculate the cosine similarity between queries, discarding those with a high similarity score above 0.9. The threshold is determined by manual inspection. Lastly, to further enhance task diversity, we used a diverse user pool by retaining only the last conversation for each unique device, thus removing tasks from the same user that might require similar underlying skills.

Difficulty annotation. To identify challenging tasks that can distinguish the performance of different LLMs, we used GPT-4-Turbo [26], Claude-3-Sonnet, and Opus [2] to analyze the required background knowledge and reasoning capabilities for each task. These models assigned a difficulty rating on a five-point scale (from "very easy," "easy," "medium," "hard," to "very hard"). Tasks rated as "very easy" or "easy" by all models were excluded. From the remaining pool, we randomly sampled 1,500 tasks to ensure that the distribution of task categories is similar to the original dataset.

Human annotation. To improve the quality of selected tasks, human annotation was used for quality control. We first used GPT-4-Turbo to summarize the intent of each query. These summaries were then reviewed to remove nonsensical tasks. Finally, we retained 1,024 tasks for WILDBENCH .

Dynamic updates and data leakage prevention. WILDBENCH is designed to be a dynamic benchmark that is updated regularly to reflect new types of user interactions. In fact, we have already released two versions of the benchmark (V1 in 2024 March and V2 in 2024 May), with similar curation process but on different iterations of WildChat data. To prevent potential data leakage for LLMs that use WildChat as part of their training or alignment, we coordinated with the WildChat team to ensure that the tasks we sample will not be publicly available in the WildChat dataset.

2.2 WILDBENCH Statistics

To better understand the composition of our evaluation, we analyze basic statistics and task categories.

[1]WildChat is released under the AI2 ImpACT license.

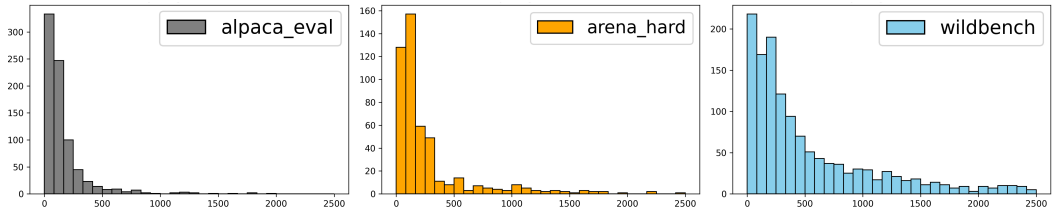


Figure 2: Distribution of query lengths in AlpacaEval, ArenaHard, and WildBench.

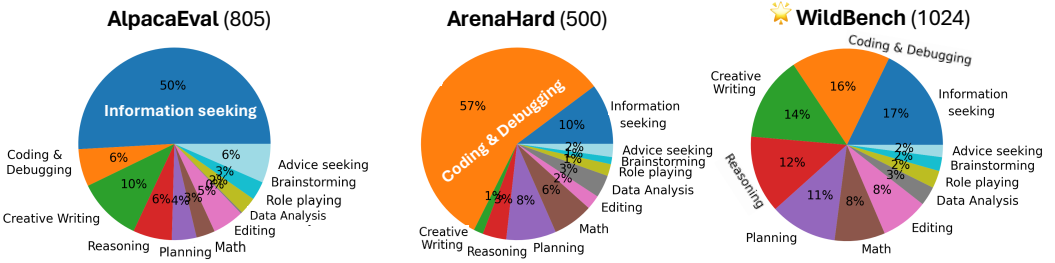


Figure 3: Distribution of task categories in AlpacaEval, ArenaHard, and WildBench.

Basic statistics. Table 1 compares the statistics of WILDBENCH to existing benchmarks AlpacaEval [15, 6], MT-Bench [38], and ArenaHard [14]. Among these benchmarks, only ArenaHard and WILDBENCH are sourced from user queries in the wild ("RealUser"), rather than being curated by experts or through crowdsourcing. The difference between ArenaHard and our WildBench is that our data distribution aligns with real users' task categories, rather than overly focusing on coding and debugging as ArenaHard does.

Long-context tasks. WILDBENCH includes conversation histories of up to four turns per conversation, reflecting complex and extended user interactions that are facilitated by recent advancements in LLMs, with over 20% of conversations having more than two or more turns as shown in Figure 6. Additionally, as shown in Figure 2, WILDBENCH has longer query lengths, attributable to the extensive context provided by real user interactions captured in the dataset. This is because that GPT-4-Turbo, one of the chatbots behind WildChat, supports up to 128K context tokens and 4K output tokens. This capability exemplifies the importance of a dynamic, in-the-wild benchmark: as models evolve, they unlock new user applications. Thanks to these realistic user activities, WILDBENCH is a more suitable benchmark for testing the long-context problem solving abilities of LLMs.

Task categories. To enable a fine-grained analysis of LLM capabilities across varied tasks, we categorize the tasks into 12 categories based on previous analysis of ShareGPT queries [27] and our intent annotation of the tasks.[2] The distribution of the task categories is shown in Figure 3. In this figure, we also compare to AlpacaEval and ArenaHard. Notably, WILDBENCH is more balanced compared to AlpacaEval and ArenaHard, which have over 50% of their tasks in Information seeking and Coding & Debugging categories, respectively.

3 Automatic Evaluation with WILDBENCH

In this section, we introduce the evaluation process of LLMs using WILDBENCH. We first explain how we generate a checklist for each test query to enhance interpretability and reduce evaluation ambiguity in WILDBENCH. Then, we introduce two automatic metrics: WILDBENCH-Score and WILDBENCH-Reward. Finally, we discuss how we mitigate the length bias in the evaluation process.

3.1 Instance-Specific Checklists

Powerful LLMs have been widely used as judges to evaluate the quality of LLM outputs in many automatic evaluation methods, such as AlpacaEval [15]. However, even asking humans to judge which of the given two model outputs is better can be subjective and ambiguous. Moreover, such judgements

[2]Detailed descriptions about the 12 task categories are shown in Appendix E.

provide limited information about the quality of the models. Without a constant, interpretable, and comprehensive evaluation standard, the results can be noisy and hard to interpret.

To address this issue, we generate a checklist for each test query in WILDBENCH to comprehensively evaluate the responses of different models. The checklist consists of 5-10 questions that are designed to be interpretable and easy to verify. We combine the responses of GPT-4-Turbo and Claude-3-Opus to finalize the checklists, thereby mitigating the bias of using a single LLM as the evaluator. These checklists have been manually reviewed and are used as part of the prompts for LLM judges to evaluate the responses of different models. An example of the checklist can be found in Figure 1.

3.2 Pairwise Evaluation with WB-Reward Metric

WB-Reward is based on pairwise evaluation, which uses a GPT-4-Turbo judge to compare the responses of two LLMs to determine which one performs better on a given task, using a structured checklist to guide the comparison. This metric provides straightforward comparisons among models and the intermediate outcomes of win/lose rates are easy to interpret.

Step-by-step evaluation process. In Figure 1, we detail the step-by-step evaluation process for pairwise comparison. First, we provide a chain of evaluation questions to guide the LLM judge to analyze the user query and the conversation history. The LLM then evaluates the two responses and also analyze where and why one is better than the other. Finally, we ask the LLM to make a final judgment on which response is better and why. This method is inspired by the evaluation process in human evaluation, where human judges are asked to provide detailed feedback on the quality of the responses before making a final decision. The full evaluation prompt can be found at Appendix A

WB-Reward metric. To compute the WB-Reward for a test model X against a baseline model Y, we assign rewards based on the comparison result: +1 if X is much better than Y, +0.5 if X is slightly better than Y, 0 for a tie, -0.5 for X is slightly worse than Y, and -1 for X is much worse than Y.

Baseline LLMs for pairwise evaluation. Using a single baseline model for pairwise evaluation can lead to noisy and biased evaluations. To mitigate this issue, we use three baseline models (GPT-4-Turbo-0429, Claude-3-Haiku, and Llama-2-70B-chat [34]) to compute the rewards for each model. Our metric WB-Reward (Mix) is the average of the rewards from these three baselines on 1024 examples, providing a more robust performance evaluation on WILDBENCH.

Mitigating length bias with a margin for ties. Previous studies have shown that LLM judges tend to prefer longer responses [6]. To mitigate this bias, we propose a simple and intuitive length penalty method. If the winning response is longer than the losing one by a certain threshold (K characters), we convert Slightly Win/Slightly Lose to a Tie. K can be customized via our leaderboard web-page for personalized configuration. Setting $K = \infty$ will disable the length penalty.

3.3 Individual Evaluation with WB-Score Metric

Although pairwise evaluation provides a direct comparison between LLMs, it is usually more expensive and time-consuming than grading each individual LLM generation. To individually evaluate the performance of each model on WILDBENCH, we prompt GPT-4-Turbo to assign a score from 1 to 10 for each model's response. The full evaluation prompt can be found at Appendix B.

Score definition. To ensure a stable and consistent evaluation, we ask GPT-4-Turbo to evaluate the quality of each response based on the checklist and provide detailed strengths and weakness of each output before giving a score from 1 to 10. The scores are defined as follows:

- Score 1–2: The response is very poor and does not make sense at all.
- Score 3–4: The response is poor and does not help the user solve the problem meaningfully.
- Score 5–6: The response is fair but has issues (e.g., factual errors, hallucinations, missing key information).
- Score 7–8: The response is good but could be improved.
- Score 9–10: The response is perfect and provides helpful information to solve the problem.

Score rescaling. The WILDBENCH-Score is calculated as the average of the scores on all examples tested, where each score is first subtracted by 5 and then multiplied by 2 (i.e., $S' = (S - 5) \times 2$). A score of 5 represents a borderline acceptable response, so this rescaling can help to better differentiate the performance of models that can effectively solve the tasks.

Table 2: Evaluation results of LLMs using WILDBENCH and other benchmarks. Please refer to Figure 5 and our website linked here to view and interact with the full results.

	Model names	WB-Reward (no length penalty) Mix	◉GPT4T	◉Haiku	◉Llama2	WB-Score	Arena Elo	Arena-Hard	AlpacaEval2 LC	AlpacaEval2 WR
1	GPT-4o-0513 🔒	35.7	1.5	46.3	59.3	65.3	1293	-	57.5	51.3
2	◉ GPT-4-Turbo-0409 🔒	34.6	0	45.3	58.4	64.7	1251	82.6	55.0	46.1
3	GPT-4-Turbo-0125 🔒	29.9	-4.4	38.8	55.2	63.3	1239	78.0	-	-
4	Gemini-1.5-Pro 🔒	27.8	-4.4	37.9	50	55.7	-	-	-	-
5	Llama-3-70B-Inst	21	-19	31.9	50.2	60.4	1213	41.1	34.4	33.2
6	Claude 3 Opus 🔒	20.1	-20.4	34.3	46.3	63.1	1232	60.4	40.5	29.1
7	Gemini-1.5-Flash 🔒	17.4	-16.6	26.3	42.5	53.1	-	-	-	-
8	Yi-1.5-34B-Chat	16.8	-18.3	24.1	44.5	57.8	-	-	-	-
10	Llama3-Inst-8B-SimPO	14	-22.5	18.9	45.7	53.9	-	33.8	44.7	40.5
13	Claude 3 Sonnet 🔒	7.2	-31.6	19.4	33.9	55.5	1187	46.8	34.9	25.6
14	Qwen1.5-72B-Chat	4.4	-34.8	13.1	34.7	56.5	1143	36.1	36.6	26.5
17	Command-R-Plus 🔒	0.4	-36.3	7.4	30.2	51.4	1155	33.1	-	-
20	◉ Claude 3 Haiku 🔒	-8.5	-46.9	0	21.4	50.4	1169	41.5	-	-
21	Mistral-Large 🔒	-10.5	-48.1	-4	20.5	54.2	1158	37.7	32.7	21.4
23	StarlingLM-7B-beta	-11.9	-48.7	-5	18	46.8	1111	23.0	-	-
24	Llama-3-8B-Inst	-14.6	-49.8	-9.7	15.7	45.7	1144	20.6	22.9	22.6
25	Command-R 🔒	-16	-48.4	-12.7	13.1	45.7	1106	17.0	-	-
26	Mixtral-8x7B-Inst	-18.8	-53.4	-13.5	10.4	47.8	1114	23.4	23.7	18.3
27	DBRX Inst	-21.6	-57.3	-16.3	8.7	48.9	1106	23.9	25.4	18.4
29	Yi-1.5-6B-Chat	-24.3	-55	-19.9	2.1	39.6	-	-	-	-
30	Mistral-7B-Inst-v0.2	-25	-58.1	-22.4	5.5	43.4	1071	-	17.1	14.7
32	Tulu-2-dpo-70b	-25.4	-59.3	-20.3	3.3	45.2	1099	15.0	21.2	16.0
33	◉ Llama-2-70B-chat	-26.8	-56.9	-23.6	0	39.2	1070	11.6	14.7	13.9
34	Qwen1.5-7B-Chat	-27	-57.7	-23	-0.2	40	1059	-	14.7	11.8
35	Phi-3-medium-128k	-33.3	-66.4	-30	-3.6	42.1	-	-	-	-
36	GPT-3.5-turbo-0125	-33.5	-66.3	-30	-4.1	42.1	1105	23.3	-	-
38	Llama-2-7B-chat	-48	-71.8	-44.6	-27.8	27.6	1012	4.6	5.4	5.0
39	Gemma-7B-it	-57	-78.4	-55.8	-36.8	23.9	1047	7.5	10.4	6.9
40	Gemma-2B-it	-74.1	-87.8	-73.6	-60.8	6.2	980	3.0	5.4	3.4

4 Results and Analysis

In this section, we analyze the performance of different models on WILDBENCH. We first present the leaderboard analysis, then examine the length bias issue in the evaluation process, and finally discuss the correlation between WILDBENCH-Score and ChatbotArena Elo rating.

Leaderboard features. In Table 2, we present a subset of the results from our live leaderboard on Hugging Face.[3] For the most up-to-date results and more interactive features, such as customizing length penalties and viewing the detailed task-wise performance of each model, please refer to our live leaderboard. Our live leaderboard also supports exploring data and comparing model outputs side by side to understand the strengths and weaknesses of each model.

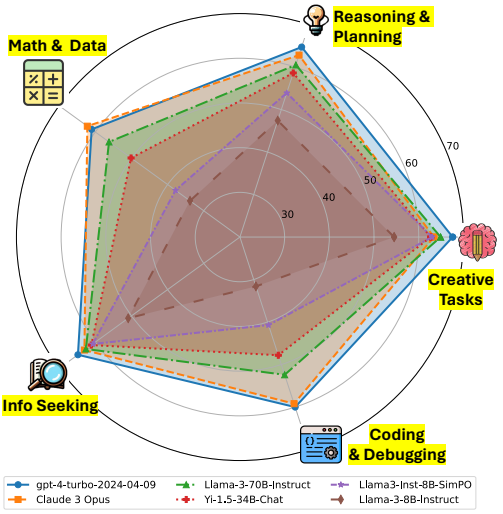


Figure 4: Performance breakdown by task category of 6 models on WILDBENCH.

By using three baseline models of varying performance levels (GPT-4-Turbo > Claude 3 Haiku > Llama-2-70B-chat), we observe that the tested models can be naturally grouped into three tiers based on their performance. Tier 1 models outperform

[3] https://huggingface.co/spaces/allenai/WildBench

Claude 3 Haiku, Tier 2 models outperform Llama-2-70B-chat but are worse than Claude 3 Haiku, and Tier 3 models are worse than Llama-2-70B-chat.

4.1 Leaderboard Analysis

Where are the gaps between models? A unique feature of the WILDBENCH leaderboard is the ability to compare models across different task categories, which enables us to identify the strengths and weaknesses of each model on different types of tasks. In Figure 4, we select a set of popular models for analysis: Llama-3-8B-Inst [22], Llama-3-8B-Inst-SimPO [20], Yi-1.5-34B-chat [1], Llama-3-70B-Inst, GPT-4-Turbo-0409, and Claude 3 Opus. We show their performance in WB-Score across five task categories (merged from the 12 categories shown in Figure 3). Larger models like GPT-4-Turbo-0409 and Claude 3 Opus perform well across all task categories, while open LLMs like Llama-3-8B-Inst and Yi-1.5-34B-chat show weaker performance on coding and math-related tasks.

Will a 8B model outperform a 70B model? On the AlpacaEval-2.0 leaderboard, Llama-3-8B-Inst-SimPO (LC=44.7%) significantly outperforms Llama-3-70B-Inst (LC=34.4%) [21], which is surprising and differs from our results. As shown in both Table 2 and Figure 4, our results indicate that Llama-3-8B-Inst-SimPO is generally still worse than Yi-34B-chat and Llama-3-70B-Inst. However, on information-seeking and creative tasks, Llama-3-8B-Inst-SimPO performs comparably to Llama-3-70B-Inst. Thus, we believe AlpacaEval's evaluation results underestimate the performance of Llama-3-70B-Inst due to task selection bias in addition to the weakness of their evaluation prompting method. While the performance of Llama-3-8B-Inst-SimPO is not as good as it seems on AlpacaEval-2.0, it is indeed the best 8B model in our evaluation and outperforms some other larger models. Interestingly, Llama-3-8B-Inst-SimPO consistently improves the performance of Llama-3-8B-Inst on all task categories, resulting in a similar shape on the radar plot in Figure 4.

Are longer responses always better? WILDBENCH is robust to length bias. For example, Llama-2-70B-chat and Llama-3-70B-Inst have similar output lengths (2965 vs 2983 chars), yet Llama-3-70B-Inst ranks 5th while Llama-2-70B-chat ranks 33rd on the leaderboard of 40 models. Additionally, Yi-1.5-6B's output length is the 4th longest among the 40 models (3322 characters), but it ranks 29th on the leaderboard. This suggests that the WILDBENCH evaluation is not biased towards longer responses, with response quality being the most important factor in the evaluation process. Additionally, we use a length penalty to ensure that longer responses are not always favored, and users can customize the length penalty to adjust the trade-off between response length and quality according to their needs. This feature is available on our live leaderboard and is illustrated in Figure 5.

4.2 Correlation to Human Judgement via ChatbotArena Elo Rating

To analyze how well WILDBENCH evaluation correlates with human judgment, we compare our results to the ChatbotArena Elo rating generated by large-scale online human evaluations. Focusing on hard prompts, we use the Elo ratings from the Hard-English version released on May 20, 2024.

We compare our WB-Reward and WB-Score with three other metrics: AlpacaEval winrate (WR), length-controlled winrate (LC), and ArenaHard scores. We use three correlation metrics: Pearson correlation (P-Cor), Spearman correlation (S-Cor), and Kendall's tau correlation (K-Cor). To ensure a fair comparison, we consider all models that have all four metrics available in Table 2, which results in 14 models. To distinguish the top-performing models, we also consider the top 6 models, denoting their correlation metrics as $\text{P-Cor}_{\text{top}}$, and $\text{P-Cor}_{\text{all}}$ respectively. The reason why we care about the correlation on top-ranking models is that models released in the future are likely to be competed with the top models, so the Pearson correlation in this range is more important from the perspective of predicting the future application of a metric. The analysis results are shown in Table 3.

Both WB-Reward and WB-Score show strong correlations with the human-based Elo rating, particularly for the top-performing models, achieving the best correlation among all other automatic metrics. Among using different baseline models for pairwise evaluation, we find that using Haiku as the baseline model yields the best correlation. These results suggest that the WILDBENCH evaluation correlates well with human judgment in ranking model performance as an automatic metric.

Table 3: Correlation with Chatbot ArenaElo Elo (Hard-En-240520) of alignment benchmarks.

Metric	P-Cor$_{top}$	P-Cor$_{all}$	S-Cor$_{all}$	K-Cor$_{all}$	Metric	P-Cor$_{top}$	P-Cor$_{all}$	S-Cor$_{all}$
ArenaElo (Hard-En)	1.000	1.000	1.000	1.000	Avg Length	0.472	0.554	0.376
Arena-Hard	*0.909*	0.925	*0.965*	*0.890*	WB-Reward$^{llama}_{\infty}$	0.976	0.965	0.965
AlpacaEval2-LC	0.892	0.951	0.924	0.818	WB-Reward$^{gpt4t}_{\infty}$	0.974	0.961	0.965
AlpacaEval2	0.865	*0.952*	0.960	0.868	WB-Reward$^{haiku}_{\infty}$	**0.985**	**0.974**	**0.982**
WB-Score	0.955	0.940	0.943	0.846	WB-Reward$^{llama}_{500}$	0.977	0.969	0.961
WB-Reward$^{mix}_{\infty}$	**0.984**	0.973	**0.978**	**0.912**	WB-Reward$^{gpt4t}_{500}$	**0.992**	0.973	0.969
WB-Reward$^{mix}_{500}$	**0.984**	**0.976**	0.974	**0.912**	WB-Reward$^{haiku}_{500}$	0.973	**0.976**	**0.974**

5 Related Works

Close-ended benchmarks. Close-ended benchmarks typically consist of multiple-choice questions and have been widely used to evaluate LLMs [3]. For example, MMLU [9] includes multi-choice questions across various subject areas. Its variants include CMMLU [13] for Chinese, KMMLU [32] for Korean, and MMLU-Pro [36] for more challenging evaluation. GPQA [31] is another close-ended benchmark designed to be challenging even for humans with internet access. Specialized benchmarks with ground-truth answers, such as GSM8K [5] and MATH [10], also fall into this category. While these benchmarks focus on close-form answers, our work evaluates LLMs' ability to generate free-form responses and engage in conversations with users.

Expert-curated and crowdsourced data. Several open-ended generation benchmarks rely on data curated by human experts or crowdsourcing workers. For instance, MT-Bench [38] manually creates examples for predefined categories. AlpacaEval [15] is based on author-written examples [7, 33, 35], which primarily consists of simple instructions such as rewriting tasks.

In-the-wild data. A key feature of our work is that its underlying data is sourced from real-world use cases, ensuring alignment with actual LLM use cases. Notable benchmarks using real-world data include ChatbotArena [38, 4], where users input their questions and choose the better response from two LLMs. However, ChatbotArena relies on extensive human feedback. ArenaHard [14] is another work that selects user queries from ChatbotArena to construct a benchmark for automatic evaluation.

Evaluation methods. Evaluating open-ended generation poses challenges due to the lack of a single valid ground truth. Human evaluation, though reliable, is expensive and time-consuming. To reduce costs and enable fast evaluation, powerful LLMs are often used as judges, as seen in benchmarks like MT-Bench, AlpacaEval, ArenaHard, and our own. Evaluation methods include single-system grading, which assigns scores to individual outputs, and pairwise comparisons, which compare outputs of two systems to compute win rates. Pairwise comparisons, while more expensive, can highlight subtle differences across systems [38]. To mitigate self-selection bias where an LLM prefers its own outputs [28], we use checklists generated from multiple LLMs, similar to InfoBench [29]. In addition, we ask LLM judges generate structured explanations that enable human verification for further calibration, inspired by Just-Eval [17].

Data leakage prevention. Publicly available benchmarks risk contamination from LLMs trained on such data. GPQA includes a special string to help LLM developers filter out its data [31], yet indirect leakage through cited examples remains possible. To mitigate this, we reserve a subset of WildChat that is never released publicly, similar to the SEAL benchmark[4], which keeps its expert-curated evaluation data private. However, WILDBENCH provides a public validation set and details the benchmark construction process for greater transparency.

Other dimensions for evaluation. While our focus is on evaluating LLM capabilities, other evaluation dimensions, such as safety [19], fairness [8], agentic planning [18, 23, 16], and hallucination detection [24, 25, 11], are equally important.

[4] https://scale.com/leaderboard

6 Limitations

Language scope. Despite the diversity of task categories included in WILDBENCH, the current version focuses on English. This limitation excludes evaluations of LLMs in other languages.

Data leakage concerns. To generate our leaderboard, we must send test data through APIs for models that do not have open-source weights. This process risks data leakage, as the test data might be inadvertently incorporated into model training data in the future. To alleviate this risk, we plan to continuously update the benchmark data. Thanks to the contribution of the WildChat project, there is always new user data.

Bias in source data. The data used in WILDBENCH is sourced from the WildChat dataset [37], which reflects the demographic distribution of its users. Consequently, any biases present in this user base are inherited by WILDBENCH. This includes potential biases in user demographics, interests, and interaction styles.

Multi-turn query biases. For multi-turn queries, responses for previous turns are generated by GPT-4, as we cannot counterfactually predict what users would say next if responses were generated by the model under evaluation.

7 Conclusion and Future Directions

In this work, we introduced WILDBENCH, a benchmark designed to evaluate LLMs using real-world user queries. By continuously updating the benchmark with new examples, WILDBENCH strives to remain relevant and reflective of the evolving capabilities of LLMs. A unique feature of WILDBENCH data is the nature of in-the-wild user queries with natural task distribution. To evaluate LLM performance using the collected data, we introduced a CoT-like LLM-as-judge method to improve the interpretability of evaluations and reduce ambiguity. We also incorporated a length penalty method to mitigate the length bias in LLM-as-judge evaluations. Experiments show that our primary metrics, WB-Reward and WB-Score, have very strong correlations with human judgments, surpassing existing metrics such as AlpacaEval 2 and ArenaHard.

We present extensive experiments and analyses, showcasing the performance of a wide range of 40 LLMs, including both proprietary and public ones, on the WILDBENCH benchmark. By providing a detailed breakdown of scores across different task categories, WILDBENCH offers insights on the strengths and weaknesses of different models. By introducing WILDBENCH, we aim to provide a realistic, dynamic, and contamination-resilient evaluation framework that accurately reflects the capabilities of LLMs. Our leaderboard at has been visited 20K times since its launch, and we will actively maintain the project for continually evaluating new LLMs with unseen tasks over time.

References

[1] 01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.ai, 2024.

[2] Anthropic. The claude 3 model family: Opus, sonnet, haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf, 2024.

[3] The BigBench authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv*, abs/2206.04615, 2022.

[4] 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.

[5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

[7] 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. *arXiv preprint arXiv:2305.14387*, 2023.

[8] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. Bias and fairness in large language models: A survey, 2024.

[9] 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.

[10] 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.

[11] Giwon Hong, Aryo Pradipta Gema, Rohit Saxena, Xiaotang Du, Ping Nie, Yu Zhao, Laura Perez-Beltrachini, Max Ryabinin, Xuanli He, Clémentine Fourrier, and Pasquale Minervini. The hallucinations leaderboard – an open effort to measure hallucinations in large language models, 2024.

[12] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *ArXiv*, abs/2205.11916, 2022.

[13] 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.

[14] 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.

[15] 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.

[16] Bill Yuchen Lin, Chengsong Huang, Qian Liu, Wenda Gu, Sam Sommerer, and Xiang Ren. On grounded planning for embodied tasks with language models. *ArXiv*, abs/2209.00465, 2022.

[17] Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Raghavi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. *ArXiv*, abs/2312.01552, 2023.

[18] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Yuxian Gu, Hangliang Ding, Kai Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Shengqi Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. *ArXiv*, abs/2308.03688, 2023.

[19] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal, 2024.

[20] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward, 2024.

[21] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. 2024.

[22] Meta. Introducing Meta Llama 3: The most capable openly available LLM to date. https://ai.meta.com/blog/meta-llama-3/, 2023.

[23] Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann André LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. *ArXiv*, abs/2311.12983, 2023.

[24] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023.

[25] Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. Fine-grained hallucination detection and editing for language models, 2024.

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

[27] Siru Ouyang, Shuohang Wang, Yang Liu, Ming Zhong, Yizhu Jiao, Dan Iter, Reid Pryzant, Chenguang Zhu, Heng Ji, and Jiawei Han. The shifted and the overlooked: A task-oriented investigation of user-GPT interactions. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[28] Arjun Panickssery, Samuel R. Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations, 2024.

[29] Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. Infobench: Evaluating instruction following ability in large language models, 2024.

[30] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[31] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023.

[32] Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. Kmmlu: Measuring massive multitask language understanding in korean. *arXiv preprint arXiv:2402.11548*, 2024.

[33] 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.

[34] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[35] 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.

[36] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024.

[37] Wenting Zhao, Xiang Ren, John Frederick Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. 2024.

[38] 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.

A Prompt Template for Pairwise Evaluation Metric WB-Reward

The prompt template for pairwise evaluation is shown below. It can be divided into three sections: the first section provides the high-level instruction, the task to be tested, and two model outputs; the second section specifies the checklist and the rules; and the last section instructs the LLM judge to follow the step-by-step evaluation process as detailed in Section 3.2

```
# Instruction
You are an expert evaluator. Your task is to evaluate the quality of the responses
↪  generated by two AI models. We will provide you with the user query and a pair
↪  of AI-generated responses (Response A and B). You should first read the user
↪  query and the conversation history carefully for analyzing the task, and then
↪  evaluate the quality of the responses based on and rules provided below.

# Conversation between User and AI

## History
<|begin_of_history|>
{$history}
<|end_of_history|>

## Current User Query
<|begin_of_query|>
{$user_query}
<|end_of_query|>

## Response A
<|begin_of_response_A|>
{$candidate_A}
<|end_of_response_A|>

## Response B
<|begin_of_response_B|>
{$candidate_B}
<|end_of_response_B|>
```

```
# Evaluation

## Checklist

<|begin_of_checklist|>
{$checklist}
<|end_of_checklist|>

Please use this checklist to guide your evaluation, but do not limit your
↪  assessment to the checklist.

## Rules

You should compare the above two responses based on your analysis of the user
↪  queries and the conversation history. You should first write down your
↪  analysis and the checklist that you used for the evaluation, and then provide
↪  your assessment according to the checklist. There are five choices to give
↪  your final assessment: ["A++", "A+", "A=B", "B+", "B++"], which correspond to
↪  the following meanings:

- `A++`: Response A is much better than Response B.
- `A+`: Response A is only slightly better than Response B.
- `A=B`: Response A and B are of the same quality. Please use this choice
↪  sparingly.
- `B+`: Response B is only slightly better than Response A.
- `B++`: Response B is much better than Response A.
```

```
## Output Format
First, please output your analysis for each model response, and then summarize
↪  your assessment to three aspects: "reason A=B", "reason A>B", and "reason
↪  B>A", and finally make your choice for the final assessment.

Please provide your evaluation results in the following json format by filling in
↪  the placeholders in []:
```
{
    "analysis of A": "[analysis of Response A]",
    "analysis of B": "[analysis of Response B]",
    "reason of A=B": "[where Response A and B perform equally well]",
    "reason of A>B": "[where Response A is better than Response B]",
    "reason of B>A": "[where Response B is better than Response A]",
    "choice": "[A++ or A+ or A=B or B+ or B++]",
}
```
```

B Prompt Template for Individual Evaluation Metric WB-Score

The prompt template for individual evaluation is shown below. It can be similarly divided into three sections: the first section provides the high-level instruction, the task to be tested, and the model output; the second section specifies the checklist and the rules; and the last section instructs the LLM judge to follow the step-by-step evaluation process as detailed in Section 3.3.

```
# Instruction

You are an expert evaluator. Your task is to evaluate the quality of the responses
↪  generated by AI models.
We will provide you with the user query and an AI-generated responses.
You should first read the user query and the conversation history carefully for
↪  analyzing the task, and then evaluate the quality of the responses based on
↪  and rules provided below.

# Conversation between User and AI

## History
<|begin_of_history|>

{$history}

<|end_of_history|>

## Current User Query
<|begin_of_query|>

{$user_query}

<|end_of_query|>

## AI Response
<|begin_of_response|>

{$model_output}

<|end_of_response|>
```

```
# Evaluation

## Checklist

<|begin_of_checklist|>

{$checklist}

<|end_of_checklist|>

Please use this checklist to guide your evaluation, but do not limit your
↪  assessment to the checklist.

## Rules

You should compare the above response based on your analysis of the user queries
↪  and the conversation history.
You should first write down your analysis and the checklist that you used for the
↪  evaluation, and then provide your assessment according to the checklist.
The scores are in the range of 1~10, where 1 means the response is very poor and
↪  10 means the response is perfect.
Here are more detailed criteria for the scores:

- Score 1~2: The response is very poor and does not make sense at all.
- Score 3~4: The response is poor and does help user solve the problem in a
↪  meaningful way.
- Score 5~6: The response is fair but has some issues (e.g., factual errors,
↪  hallucinations, missing key information).
- Score 7~8: The response is good enough but could be improved in some ways.
- Score 9~10: The response is perfect and provides helpful information that can
↪  help user solve the problem.
```

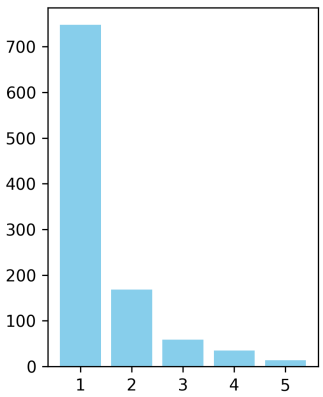
```
## Output Format
First, please output your analysis for each model response, and then summarize
↪  your assessment to three aspects: "reason A=B", "reason A>B", and "reason
↪  B>A", and finally make your choice for the final assessment.

Please provide your evaluation results in the following json format by filling in
↪  the placeholders in []:
```
{
    "strengths": "[analysis for the strengths of the response]",
    "weaknesses": "[analysis for the weaknesses of the response]",
    "score": "[1~10]"
}
```
```

C More Information on WILDBENCH Data

The distribution of the number of turns in WILDBENCH can be found in Figure 6. The dataset documentation, metadata, and the public subset of WILDBENCH can be found at https://huggingface.co/datasets/allenai/WildBench/viewer/v2. We release the data under AI2's ImpACT license as a low-risk artifact[5], and we bear all responsibility in case of rights violations. We will ensure that the dataset will be available for a long time and maintain the data by continuously updating it.

[5] https://allenai.org/licenses/impact-lr

Figure 6: Distribution of the number of turns in WildBench.



D More Information on WILDBENCH Evaluation

Our evaluation results on the public subset of WILDBENCH can be reproduced using evaluation scripts available at https://github.com/allenai/WildBench/tree/main. We have included generation script for each model under the folder https://github.com/allenai/WildBench/tree/main/scripts, and the scripts for evaluating generations can be found at https://github.com/allenai/WildBench/tree/main/evaluation.

E Task Categories

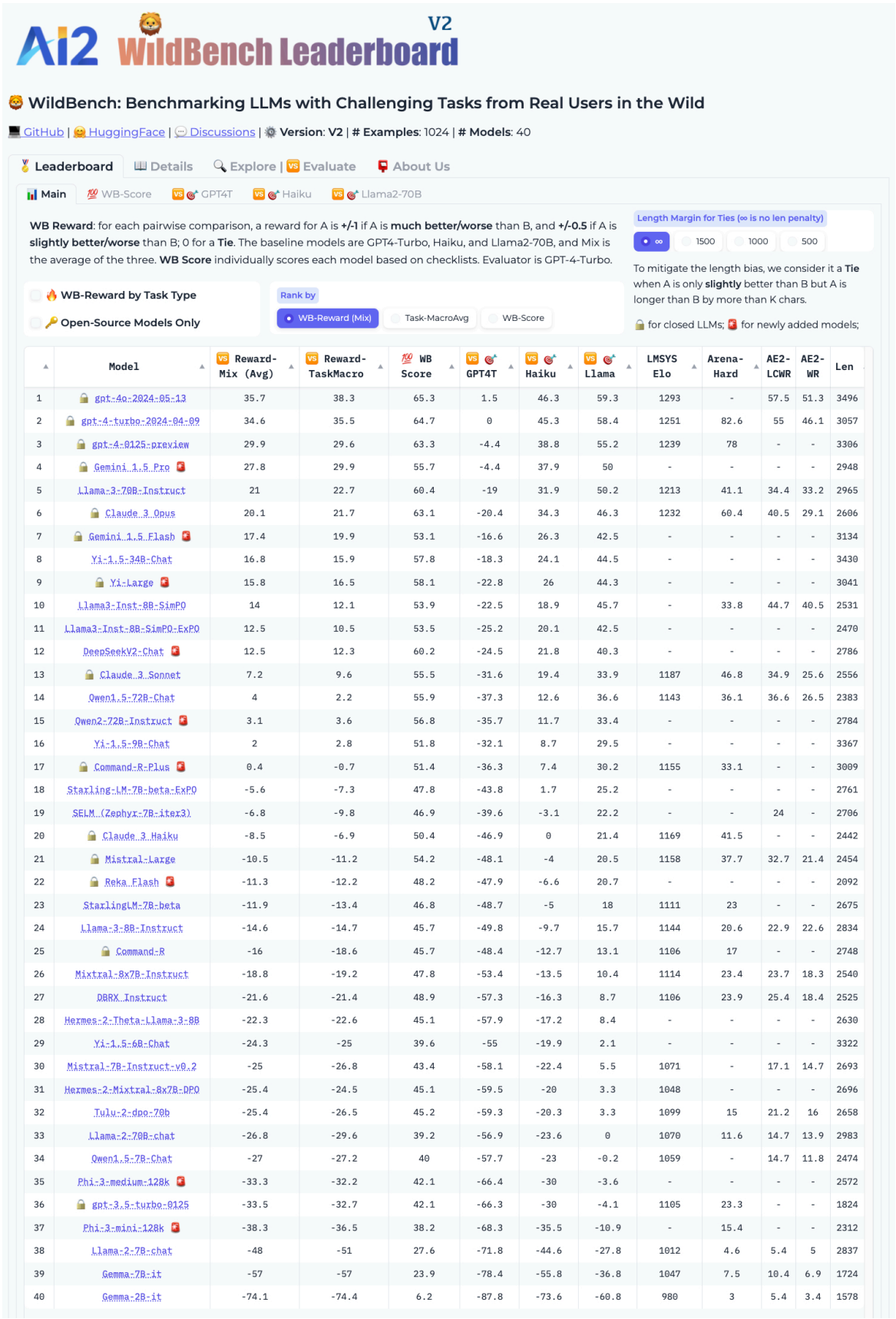
In Section 2.2 we mentioned that tasks are cateogrized into 12 categories to enable fine-grained analysis of LLM capabilities. The definition of these task categoriezs are as follows.

- **Information seeking** - Users ask for specific information or facts about various topics.
- **Reasoning** - Queries require logical thinking, problem-solving, or processing of complex ideas.
- **Planning** - Users need assistance in creating plans or strategies for activities and projects.
- **Editing** - Involves editing, rephrasing, proofreading, or other tasks related to the composition of general written content.
- **Coding & Debugging** - Users seek help with writing, reviewing, or fixing code in programming.
- **Math** - Queries related to mathematical concepts, problems, and calculations.
- **Role playing** - Users engage in scenarios requiring ChatGPT to adopt a character or persona.
- **Data Analysis** - Requests involve interpreting data, statistics, or performing analytical tasks.
- **Creative Writing** - Users seek assistance with crafting stories, poems, or other creative texts.
- **Advice seeking** - Users ask for recommendations or guidance on various personal or professional issues.
- **Brainstorming** - Involves generating ideas, creative thinking, or exploring possibilities.
- **Others** - Any queries that do not fit into the above categories or are of a miscellaneous nature.

We consolidate the original categories into five major groups for easier task-wise analysis. Specifically, we combine "Information seeking" and "Advice seeking" into "Info Seeking"; "Math" and "Data Analysis" into "Math & Data"; and "Reasoning" and "Planning" into "Reasoning & Planning." The remaining types are grouped under "Creative Tasks." These consolidated groups are illustrated in Figure 4.

F Full WILDBENCH Leaderboard

The full WILDBENCH leaderboard as of Jun 5, 2024 can be found in Figure 5. You can view and interact with the latest results on our leaderboard on our website at https://huggingface.co/spaces/allenai/WildBench



	Model	Reward-Mix (Avg)	Reward-TaskMacro	WB Score	GPT4T	Haiku	Llama	LMSYS Elo	Arena-Hard	AE2-LCWR	AE2-WR	Len
1	gpt-4o-2024-05-13	35.7	38.3	65.3	1.5	46.3	59.3	1293	-	57.5	51.3	3496
2	gpt-4-turbo-2024-04-09	34.6	35.5	64.7	0	45.3	58.4	1251	82.6	55	46.1	3057
3	gpt-4-0125-preview	29.9	29.6	63.3	-4.4	38.8	55.2	1239	78	-	-	3306
4	Gemini 1.5 Pro	27.8	29.9	55.7	-4.4	37.9	50	-	-	-	-	2948
5	Llama-3-70B-Instruct	21	22.7	60.4	-19	31.9	50.2	1213	41.1	34.4	33.2	2965
6	Claude 3 Opus	20.1	21.7	63.1	-20.4	34.3	46.3	1232	60.4	40.5	29.1	2606
7	Gemini 1.5 Flash	17.4	19.9	53.1	-16.6	26.3	42.5	-	-	-	-	3134
8	Yi-1.5-34B-Chat	16.8	15.9	57.8	-18.3	24.1	44.5	-	-	-	-	3430
9	Yi-Large	15.8	16.5	58.1	-22.8	26	44.3	-	-	-	-	3041
10	Llama3-Inst-8B-SimPO	14	12.1	53.9	-22.5	18.9	45.7	-	33.8	44.7	40.5	2531
11	Llama3-Inst-8B-SimPO-ExPO	12.5	10.5	53.5	-25.2	20.1	42.5	-	-	-	-	2470
12	DeepSeekV2-Chat	12.5	12.3	60.2	-24.5	21.8	40.3	-	-	-	-	2786
13	Claude 3 Sonnet	7.2	9.6	55.5	-31.6	19.4	33.9	1187	46.8	34.9	25.6	2556
14	Qwen1.5-72B-Chat	4	2.2	55.9	-37.3	12.6	36.6	1143	36.1	36.6	26.5	2383
15	Qwen2-72B-Instruct	3.1	3.6	56.8	-35.7	11.7	33.4	-	-	-	-	2784
16	Yi-1.5-9B-Chat	2	2.8	51.8	-32.1	8.7	29.5	-	-	-	-	3367
17	Command-R-Plus	0.4	-0.7	51.4	-36.3	7.4	30.2	1155	33.1	-	-	3009
18	Starling-LM-7B-beta-ExPO	-5.6	-7.3	47.8	-43.8	1.7	25.2	-	-	-	-	2761
19	SELM (Zephyr-7B-iter3)	-6.8	-9.8	46.9	-39.6	-3.1	22.2	-	-	24	-	2706
20	Claude 3 Haiku	-8.5	-6.9	50.4	-46.9	0	21.4	1169	41.5	-	-	2442
21	Mistral-Large	-10.5	-11.2	54.2	-48.1	-4	20.5	1158	37.7	32.7	21.4	2454
22	Reka Flash	-11.3	-12.2	48.2	-47.9	-6.6	20.7	-	-	-	-	2092
23	StarlingLM-7B-beta	-11.9	-13.4	46.8	-48.7	-5	18	1111	23	-	-	2675
24	Llama-3-8B-Instruct	-14.6	-14.7	45.7	-49.8	-9.7	15.7	1144	20.6	22.9	22.6	2834
25	Command-R	-16	-18.6	45.7	-48.4	-12.7	13.1	1106	17	-	-	2748
26	Mixtral-8x7B-Instruct	-18.8	-19.2	47.8	-53.4	-13.5	10.4	1114	23.4	23.7	18.3	2540
27	DBRX Instruct	-21.6	-21.4	48.9	-57.3	-16.3	8.7	1106	23.9	25.4	18.4	2525
28	Hermes-2-Theta-Llama-3-8B	-22.3	-22.6	45.1	-57.9	-17.2	8.4	-	-	-	-	2630
29	Yi-1.5-6B-Chat	-24.3	-25	39.6	-55	-19.9	2.1	-	-	-	-	3322
30	Mistral-7B-Instruct-v0.2	-25	-26.8	43.4	-58.1	-22.4	5.5	1071	-	17.1	14.7	2693
31	Hermes-2-Mixtral-8x7B-DPO	-25.4	-24.5	45.1	-59.5	-20	3.3	1048	-	-	-	2696
32	Tulu-2-dpo-70b	-25.4	-26.5	45.2	-59.3	-20.3	3.3	1099	15	21.2	16	2658
33	Llama-2-70B-chat	-26.8	-29.6	39.2	-56.9	-23.6	0	1070	11.6	14.7	13.9	2983
34	Qwen1.5-7B-Chat	-27	-27.2	40	-57.7	-23	-0.2	1059	-	14.7	11.8	2474
35	Phi-3-medium-128k	-33.3	-32.2	42.1	-66.4	-30	-3.6	-	-	-	-	2572
36	gpt-3.5-turbo-0125	-33.5	-32.7	42.1	-66.3	-30	-4.1	1105	23.3	-	-	1824
37	Phi-3-mini-128k	-38.3	-36.5	38.2	-68.3	-35.5	-10.9	-	15.4	-	-	2312
38	Llama-2-7B-chat	-48	-51	27.6	-71.8	-44.6	-27.8	1012	4.6	5.4	5	2837
39	Gemma-7B-it	-57	-57	23.9	-78.4	-55.8	-36.8	1047	7.5	10.4	6.9	1724
40	Gemma-2B-it	-74.1	-74.4	6.2	-87.8	-73.6	-60.8	980	3	5.4	3.4	1578

Figure 5: Leaderboard of WildBench (2024 Jun 5th)