

# *The Good, The Bad, and The Greedy:* Evaluation of LLMs Should Not Ignore Non-Determinism

Anonymous ACL submission

## Abstract

Current evaluations of large language models (LLMs) often overlook non-determinism, typically focusing on a single output per example. This limits our understanding of LLM performance variability in real-world applications. Our study addresses this issue by exploring key questions about the performance differences between greedy decoding and sampling, identifying benchmarks' consistency regarding non-determinism, and examining unique model behaviors. Our findings reveal and quantify significant performance gaps between greedy and sampling methods across various benchmarks, with sampling excelling in creative tasks and greedy decoding favoring deterministic tasks. We also observe consistent performance across different LLM sizes and alignment methods, noting that alignment can reduce sampling variance. Moreover, our best-of-N sampling approach demonstrates that smaller LLMs can match or surpass larger models such as GPT-4-Turbo, highlighting the untapped potential of smaller LLMs. This research shows the importance of considering non-determinism in LLM evaluations and provides insights for future LLM development and evaluation.<sup>1</sup>

## 1 Introduction

When evaluating a large language model (LLM), two common generation configurations are commonly used: greedy decoding and nucleus sampling (Holtzman et al., 2019). It's important to note that given a particular input, the same LLM may generate significantly different outputs under various decoding configurations, a phenomenon known as non-determinism in generation. However, most evaluations of LLMs are based on a single output per example. This practice is primarily due to practical considerations, as LLM inference and evaluation can be computationally expensive. Neglecting non-determinism in generation significantly limits

our comprehensive understanding of LLMs. Additionally, without reporting the standard deviation in most current LLM evaluations, it is difficult to measure the variability and dynamics of LLMs in real-world applications.

For certain capabilities such as math reasoning (Cobbe et al., 2021; Hendrycks et al., 2021) and coding, greedy generation is preferred to ensure fair comparisons. Nonetheless, it remains unclear whether there are significant differences in performance between greedy decoding and sampling. Recent investigations have also highlighted potential issues of instability in LLMs (Li et al., 2024a; Hassid et al., 2024). In a study where the best answer was selected from 256 random generations, the Llama-2-7B model achieved an impressive 97.7% accuracy in solving GSM8K questions, even surpassing GPT-4 (Li et al., 2024a). This phenomenon further underscores the enormous potential of LLMs in their non-deterministic outputs.

Herein, we aim to investigate a series of critical questions regarding the non-determinism of LLM generations, which have not been fully explored:

- **Q1:** *How does the performance gap between greedy decoding and sampling differ?*
- **Q2:** *When is greedy decoding better than sampling, and vice versa? Why?*
- **Q3:** *Which benchmark is most/least consistent with respect to non-determinism?*
- **Q4:** *Do any models possess unique patterns?*

Apart from Q1-Q4 in Sec. 3.1, we also explore the *scaling* effect on non-determinism (Sec. 3.2), the *alignment* effect on non-determinism (Sec. 3.3), and the full *potential* of LLMs (Sec. 3.4).

Our extensive results reveal these findings:

- For most benchmarks we evaluated, a notable performance gap is observed between greedy generation and the average score of multiple sampling. In certain cases, the performance ranking under different generation configurations differs.
- Sampling methods perform better on tasks de-

<sup>1</sup>Our code, data, and results will be open-sourced.

Model	AlpacaEval 2 (N=16)				Arena-Hard (N=16)				MixEval (N=16)			
	Greedy	Sample	Std.	$\Delta$	Greedy	Sample	Std.	$\Delta$	Greedy	Sample	Std.	$\Delta$
GPT-4-Turbo	49.6	50.1	0.76	2.5	80.1	75.2	1.31	3.6	89.2	88.8	0.18	0.8
Llama-3-8B-Instruct	26.8	29.2	0.88	2.8	23.5	18.4	0.71	2.7	74.6	72.5	0.25	0.9
Yi-1.5-6B-Chat	17.5	18.0	0.91	3.4	13.7	11.8	0.88	3.1	70.0	68.6	0.26	1.0
Yi-1.5-9B-Chat	23.1	24.1	0.91	3.4	32.8	27.0	1.25	4.4	74.0	72.7	0.35	1.4
Yi-1.5-34B-Chat	34.9	35.0	0.99	3.9	42.8	40.9	1.82	5.7	81.9	81.8	0.47	1.5
Qwen2-7B-Instruct	18.2	19.1	2.51	8.6	23.7	16.1	0.87	3.1	76.2	76.2	0.21	0.6
Mistral-7B-Instruct-v0.2	15.4	13.0	1.02	4.2	12.5	12.6	0.57	2.0	69.8	70.0	0.24	0.9

Model	MMLU-Redux (N=32)				GSM8K (N=128)				HumanEval (N=128)			
	Greedy	Sample	Std.	$\Delta$	Greedy	Sample	Std.	$\Delta$	Greedy	Sample	Std.	$\Delta$
GPT-4-Turbo	82.6	82.4	0.43	1.6	84.5	83.8	0.77	2.5	89.6	84.1	2.65	11.0
Llama-3-8B-Instruct	50.4	50.7	0.70	2.8	58.6	64.4	2.50	13.4	30.5	31.8	3.62	18.3
Yi-1.5-6B-Chat	48.7	49.6	0.67	2.5	74.5	73.1	0.92	4.1	48.2	35.7	4.86	19.5
Yi-1.5-9B-Chat	64.4	64.3	0.53	2.3	82.9	81.0	0.69	3.9	55.5	36.4	4.92	27.5
Yi-1.5-34B-Chat	82.6	82.2	0.34	1.1	85.4	81.7	0.56	2.9	64.6	49.3	4.08	21.4
Qwen2-7B-Instruct	61.0	61.7	0.46	2.1	83.5	72.0	1.74	11.3	67.7	48.2	4.68	27.4
Mistral-7B-Instruct-v0.2	48.7	48.4	0.49	2.2	45.9	42.0	0.99	5.1	37.8	25.9	2.52	14.0

Table 1: Results on six popular benchmarks. “Sample” and “Std.” denotes the average score and the standard deviation of “N” runs under sampling setup. “ $\Delta$ ” denotes the performance gap between the best and worst run. Scores where greedy decoding surpasses the sampling average are highlighted in green, while those lower are marked in red. The intensity of the color indicates the magnitude of the difference (best viewed in color).

manding creative writing capabilities, such as AlpacaEval, whereas deterministic tasks like those related to mathematics and coding favor greedy decoding for enhanced effectiveness.

- LLMs displayed consistent performance across different generation configurations for benchmarks with constrained output spaces, such as MMLU and MixEval. Notably, tasks involving math reasoning and code generation were most impacted by sampling variance.
- The above findings remain consistent across different sizes and families of LLMs.
- Alignment methods, e.g., DPO (Rafailov et al., 2024), can significantly reduce the sampling variance for most benchmarks.
- 7B-level LMs have the potential to outperform GPT-4-Turbo by best-of-N sampling.

## 2 Experimental Setup

**Benchmarks.** We select multiple benchmarks for our experiments, encompassing abilities of general instruction-following, knowledge, math reasoning, coding, etc. The selected benchmarks are: AlpacaEval 2 (Li et al., 2023), Arena-Hard (Li et al., 2024b), MMLU-Redux (Gema et al., 2024), MixEval (Ni et al., 2024), GSM8K (Cobbe et al., 2021), and HumanEval (Chen et al., 2021). See Appendix A for more descriptions about them.

**LLMs.** We test several open-weight LLMs, including Llama-3-Instruct (Meta, 2024), Yi-1.5-Chat (Young et al., 2024), Qwen-2-Instruct (Bai

et al., 2023), Mistral (Jiang et al., 2023a), which are widely used. A proprietary LLM, GPT-4-Turbo, is included for comparison. We also consider models of different sizes in the same family such as Qwen2 and Yi-1.5 for more analysis. To study the effect of alignment techniques, we evaluate models trained with different alignment methods, including DPO (Rafailov et al., 2024), KTO (Ethayarajh et al., 2024), SimPO (Meng et al., 2024). We use the checkpoints released by Meng et al. (2024).

**Setup.** We aim to compare the performance of LLMs under different decoding configurations. We select greedy decoding and sampling generation for the main comparison. For sampling, we set the temperature to 1.0 and top-p to 1.0. Please refer to Appendix B for more details.

## 3 Experimental Results & Analysis

In this section, we first present our results and analyze several research questions around the non-determinism of LLM generations.

### 3.1 Main Results


Our extensive experiment results are shown in Table 1. We analyze the results and answer several important research questions as follows.



Q1. How does the performance gap between greedy decoding and sampling differ?

From Table 1, we observe a consistent performance gap between greedy decoding and the sampling method. This disparity holds true across


various LLMs, whether they are proprietary or open-source, and across multiple benchmarks encompassing instruction-following, language understanding, math reasoning, and code generation. Different decoding configurations can even alter the model rankings in some cases. For example, on Arena-Hard, Qwen2-7B is slightly better than Llama-3-8B when both use greedy decoding; However, Llama-3-8B may outperform Qwen2-7B when both decode by sampling.

 Q2. When is greedy decoding better than sampling, and vice versa? Why?

Most evaluated models show higher win rates with sampling on AlpacaEval. Conversely, on benchmarks like Arena-Hard, MixEval, GSM8K, and HumanEval, greedy decoding performs better.

GSM8K and HumanEval are reasoning tasks requiring LLMs to solve specific math or coding problems with definite solutions. MixEval also follows a deterministic pattern with its ground-truth-based benchmarks. Although both AlpacaEval and Arena-Hard are open-ended information-following benchmarks, their behavior diverges significantly. As noted by Lin et al. (2024), 50% of instances in AlpacaEval are information-seeking, whereas more than 50% in Arena-Hard are related to coding and debugging. In other words, AlpacaEval demands a higher degree of creativity.

In summary: 1) For deterministic tasks, such as math and coding, greedy decoding is generally more effective. 2) For open-ended creative tasks, like information seeking and brainstorming, sampling tends to generate better responses.

 Q3. Which benchmark is most/least consistent with respect to non-determinism?

MixEval and MMLU exhibit the highest stability, either in terms of the performance gap between greedy decoding and sampling or the standard deviation across different samplings. This stability can be attributed to the constrained answer space of these benchmarks. Specifically, MMLU is structured in a multiple-choice format, and MixEval, comprising various ground-truth-based benchmarks, prompts LLMs to generate short answers, further limiting the output space.


In contrast, GSM8K and HumanEval are relatively less stable with respect to non-deterministic generations. The performance gap between the best and worst samplings can exceed 10.0 points.

Model	AlpacaEval			MMLU		
	G	S	Std.	G	S	Std.
Qwen2-0.5B-Instruct	1.1	1.7	0.77	36.4	37.0	0.70
Qwen2-1.5B-Instruct	1.9	3.3	0.88	42.6	42.1	0.68
Qwen2-7B-Instruct	18.2	19.1	2.51	61.0	61.7	0.46

Model	GSM8K			HumanEval		
	G	S	Std.	G	S	Std.
Qwen2-0.5B-Instruct	31.7	14.3	1.86	28.0	10.8	2.14
Qwen2-1.5B-Instruct	63.1	36.5	3.20	40.9	22.6	2.94
Qwen2-7B-Instruct	83.5	72.0	1.74	67.7	48.2	4.68

Table 2: Evaluation results on Qwen2-Instruct with different model sizes.

 Q4. Do the models possess distinctive characteristics?

GPT-4-Turbo shows consistent performance across multiple tasks, with a smaller performance gap between greedy decoding and sampling, as well as improved sampling quality. Some open-weight LLMs, however, exhibit unique characteristics. For example, Mistral-7B-Instruct-v0.2 displays inverse behavior on open-ended tasks like AlpacaEval and Arena-Hard when compared to other models. Similarly, Llama-3-8B-Instruct performs better by sampling than by greedy decoding on GSM8K and HumanEval, which is unlike the behavior of other models.

These observations raise intriguing questions for future research. Why do certain models exhibit inverse behavior on specific tasks? Can these unique characteristics be leveraged to develop more robust LLMs? These questions highlight the need for deeper explorations into the underlying mechanisms of LLMs. Such research could significantly enhance our understanding of how different models and training impact model behavior.

### 3.2 Scaling Effect on Non-Determinism

Some might assume that larger LMs will have lower uncertainty in decoding, leading to lower variance in performance when sampling. However, our results challenge this assumption.

We use the Yi-1.5-Chat and Qwen2-Instruct series to investigate the scaling effect. The results for the Yi-1.5 and Qwen2 series are presented in Table 1 and Table 2, respectively. Performance differences are observed across LLMs of various sizes, ranging from 0.5B to 34B parameters. The findings in Section 3.1 are consistent across different model sizes. However, no pattern related to the number of model parameters could be identified.

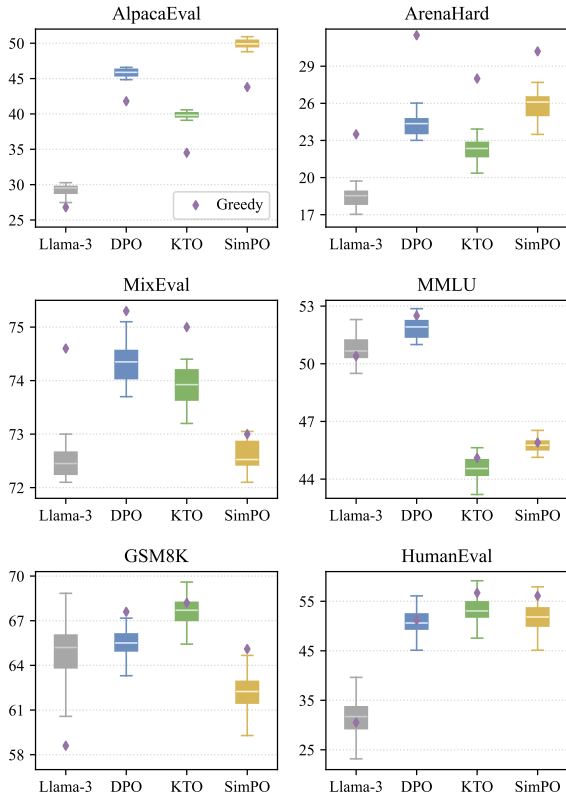


Figure 1: Alignment effects on non-determinism.

For instance, scaling parameters does not result in lower sampling variance. Notably, Qwen2-7B-Instruct shows higher variance on AlpacaEval and HumanEval compared to its smaller counterparts.

### 3.3 Alignment Effect on Non-Determinism

Alignment methods, such as DPO, enhance LLMs by learning from preference data. We evaluate the effects of alignment methods such as DPO, KTO, and SimPO, using Llama-3-8B-Instruct as the training starting point (Meng et al., 2024).

As shown in Figure 1, after applying these methods, both greedy decoding and sampling performances are affected. In several tasks, including AlpacaEval, MMLU, GSM8K, and HumanEval, a decrease in standard deviation is observed, suggesting that alignment may reduce the diversity of sampling outputs. However, it is crucial to note that not all alignment methods consistently improve model performance. For instance, KTO and SimPO lead to a performance decline in MMLU. Furthermore, SimPO’s effectiveness appears limited on the recently introduced MixEval benchmark.

### 3.4 What is the full potential of LLM?

Current evaluations of LLMs mainly assess them based on a single output per instance, which limits our understanding of their full potential. Following Jiang et al. (2023b) and Li et al. (2024a), we

Model	Setting	AE	MMLU	GSM	HE
GPT-4-Turbo	Sample-Avg	50.1	82.4	83.8	84.1
Llama-3-8B-Ins.	Sample-Avg	29.2	50.7	64.4	31.8
	Sample-Max	30.3	52.3	68.8	41.5
	Best-of-N	41.2	90.3	99.4	92.1
Yi-1.5-6B-Chat	Sample-Avg	18.0	49.6	73.1	36.4
	Sample-Max	19.5	51.0	75.0	45.7
	Best-of-N	43.3	89.6	98.4	91.5

Table 3: Potential of LLMs. “Max” denotes the best score of N runs. “Best-of-N” means we select the best response from N outputs for each example.

adopt a Best-of-N setting, selecting the best answer from N sampled responses. The results are shown in Table 3. With these results, we observe that smaller LLMs, such as **Llama-3-8B-Instruct** and **Yi-1.5-6B-Chat**, can nearly match or even surpass the performance of GPT-4-Turbo on AlpacaEval (AE), MMLU, GSM8K, and HumanEval. This finding suggests that compact-sized LLMs already exhibit robust capabilities, highlighting that a more significant challenge in alignment is to robustly decode such knowledge and reasoning paths.

Building upon these promising findings, there are two ways to further enhance the performance of smaller LLMs. Firstly, probability calibration techniques can guide LLMs towards generating superior answers with higher likelihoods. Alignment methods, specifically preference optimization (Rafailov et al., 2024), play a pivotal role in this process. Secondly, strategies for ensemble learning or selecting the best answer from multiple completions warrant attention. Reward modeling for re-ranking and fusing multiple outputs is thus a key direction (Jiang et al., 2023b). Self-consistency (Wang et al., 2022) and advanced prompting techniques (Yao et al., 2023; Lin et al., 2023), which employs heuristic selection from multiple completions, is also worth further exploration.

## 4 Conclusion & Future directions

We investigate a series of critical yet overlooked questions around non-determinism of LLM generations. After evaluating several LLMs across six commonly used benchmarks, we have answered several intriguing research questions. Further analysis also provides insights on how scaling and alignment will effect on non-determinism generation. We hope this work can enhance our comprehension of the generation methods and the widely used benchmarks. Our evaluation results can also be used for improving future research. For example, our best-of-N results can serve as a benchmark for assessing reward models (Lambert et al., 2024).

## 288 **Limitations**

289 The comparison of greedy decoding and sampling  
290 in this work reveals intriguing findings. However,  
291 it is crucial to acknowledge the limitations of our  
292 research. 1) We only conduct experiments with  
293 temperature at 0.0 and 1.0, leaving the fine-grained  
294 ablation of different temperature values on LLM  
295 performance unexplored. 2) The influence of other  
296 generation parameters besides temperature, such as  
297 repetition penalty, on LLM performance remains to  
298 be future work. 3) Our evaluation exclusively relies  
299 on off-the-shelf benchmarks, neglecting the analy-  
300 sis of other content characteristics such as language  
301 style. 4) While we showcase the remarkable poten-  
302 tial of LLMs to exhibit robust capabilities, how to  
303 incorporate methods, such as self-consistency and  
304 blender, to improve the performance of LLMs in a  
305 multiple generation setting is under-explore.

## 306 **Ethics Statement**

307 This work fully complies with the ACL Ethics Pol-  
308 icy. We declare that there are no ethical issues in  
309 this paper, to the best of our knowledge.

## 310 **References**

311 Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang,  
312 Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei  
313 Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin,  
314 Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu,  
315 Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren,  
316 Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong  
317 Tu, Peng Wang, Shijie Wang, Wei Wang, Sheng-  
318 guang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang,  
319 Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu,  
320 Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingx-  
321 uan Zhang, Yichang Zhang, Zhenru Zhang, Chang  
322 Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang  
323 Zhu. 2023. Qwen technical report. *arXiv preprint*  
324 *arXiv:2309.16609*.

325 Mark Chen, Jerry Tworek, Heewoo Jun, Qiming  
326 Yuan, Henrique Ponde de Oliveira Pinto, Jared Ka-  
327 plan, Harri Edwards, Yuri Burda, Nicholas Joseph,  
328 Greg Brockman, et al. 2021. Evaluating large  
329 language models trained on code. *arXiv preprint*  
330 *arXiv:2107.03374*.

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

336 Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff,  
337 Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model  
338 alignment as prospect theoretic optimization. *arXiv*  
339 *preprint arXiv:2402.01306*.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon  
Hong, Alessio Devoto, Alberto Carlo Maria Mancino,  
Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mo-  
hammad Reza Ghasemi Madani, et al. 2024. Are we  
done with mmlu? *arXiv preprint arXiv:2406.04127*. 340  
341 342 343 344

Michael Hassid, Tal Remez, Jonas Gehring, Roy  
Schwartz, and Yossi Adi. 2024. The larger the better?  
improved llm code-generation via budget realloca-  
tion. *arXiv preprint arXiv:2404.00725*. 345  
346 347 348

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou,  
Mantas Mazeika, Dawn Song, and Jacob Steinhardt.  
2020. Measuring massive multitask language under-  
standing. *arXiv preprint arXiv:2009.03300*. 349  
350 351 352

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul  
Arora, Steven Basart, Eric Tang, Dawn Song, and Ja-  
cob Steinhardt. 2021. Measuring mathematical prob-  
lem solving with the math dataset. *arXiv preprint*  
*arXiv:2103.03874*. 353  
354 355 356 357

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and  
Yejin Choi. 2019. The curious case of neural text  
degeneration. *arXiv preprint arXiv:1904.09751*. 358  
359 360

Albert Q Jiang, Alexandre Sablayrolles, Arthur Men-  
sch, Chris Bamford, Devendra Singh Chaplot, Diego  
de las Casas, Florian Bressand, Gianna Lengyel, Guil-  
laume Lample, Lucile Saulnier, et al. 2023a. Mistral  
7b. *arXiv preprint arXiv:2310.06825*. 361  
362 363 364 365

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b.  
[Llm-blender: Ensembling large language models  
with pairwise ranking and generative fusion](#). In *An-  
nual Meeting of the Association for Computational*  
*Linguistics*. 366  
367 368 369 370

Nathan Lambert, Valentina Pyatkin, Jacob Daniel Mor-  
rison, Lester James Validad Miranda, Bill Yuchen  
Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin  
Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and  
Hanna Hajishirzi. 2024. [Rewardbench: Evaluat-  
ing reward models for language modeling](#). *ArXiv*,  
abs/2403.13787. 371  
372 373 374 375 376 377

Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nan-  
ning Zheng, Han Hu, Zheng Zhang, and Houwen  
Peng. 2024a. Common 7b language models already  
possess strong math capabilities. *arXiv preprint*  
*arXiv:2403.04706*. 378  
379 380 381 382

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap,  
Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica.  
2024b. [From live data to high-quality benchmarks:  
The arena-hard pipeline](#). 383  
384 385 386

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori,  
Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and  
Tatsunori B. Hashimoto. 2023. AlpacaEval: An au-  
tomatic evaluator of instruction-following models.  
[https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval). 387  
388 389 390 391

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze  
Brahman, Abhilasha Ravichander, Valentina Pyatkin,  
Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2024. 392  
393 394

395 Wildbench: Benchmarking llms with challenging  
396 tasks from real users in the wild. *arXiv preprint*  
397 *arXiv:2406.04770*.

398 Bill Yuchen Lin, Yicheng Fu, Karina Yang, Prithvi-  
399 raj Ammanabrolu, Faeze Brahman, Shiyu Huang,  
400 Chandra Bhagavatula, Yejin Choi, and Xiang Ren.  
401 2023. [Swiftsage: A generative agent with fast and](#)  
402 [slow thinking for complex interactive tasks](#). *ArXiv*,  
403 [abs/2305.17390](#).

404 Yu Meng, Mengzhou Xia, and Danqi Chen.  
405 2024. [Simpo: Simple preference optimization](#)  
406 [with a reference-free reward](#). *arXiv preprint*  
407 *arXiv:2405.14734*.

408 Meta. 2024. [Introducing meta llama 3: The most capa-](#)  
409 [ble openly available llm to date](#).

410 Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng,  
411 Mahir Shah, Kabir Jain, Graham Neubig, and Yang  
412 You. 2024. [Mixeval: Deriving wisdom of the](#)  
413 [crowd from llm benchmark mixtures](#). *arXiv preprint*  
414 *arXiv:[placeholder]*.

415 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christo-  
416 pher D Manning, Stefano Ermon, and Chelsea Finn.  
417 2024. [Direct preference optimization: Your language](#)  
418 [model is secretly a reward model](#). *Advances in Neu-*  
419 *ral Information Processing Systems*, 36.

420 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le,  
421 Ed Chi, Sharan Narang, Aakanksha Chowdhery, and  
422 Denny Zhou. 2022. [Self-consistency improves chain](#)  
423 [of thought reasoning in language models](#). *arXiv*  
424 *preprint arXiv:2203.11171*.

425 Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack  
426 Hessel, Tushar Khot, Khyathi Chandu, David Wad-  
427 den, Kelsey MacMillan, Noah A Smith, Iz Beltagy,  
428 et al. 2023. [How far can camels go? exploring the](#)  
429 [state of instruction tuning on open resources](#). *Ad-*  
430 *vances in Neural Information Processing Systems*,  
431 36:74764–74786.

432 Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni,  
433 Abhranil Chandra, Shiguang Guo, Weiming Ren,  
434 Aaran Arulraj, Xuan He, Ziyang Jiang, et al. 2024.  
435 [Mmlu-pro: A more robust and challenging multi-task](#)  
436 [language understanding benchmark](#). *arXiv preprint*  
437 *arXiv:2406.01574*.

438 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran,  
439 Thomas L. Griffiths, Yuan Cao, and Karthik  
440 Narasimhan. 2023. [Tree of thoughts: Deliberate](#)  
441 [problem solving with large language models](#). *ArXiv*,  
442 [abs/2305.10601](#).

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

Dataset	Instance Num.	Sample Num.	Metric
AlpacaEval 2	805	16	LC
Arena-Hard	500	16	WR
MixEval	4000	16	Score
MMLU-Redux	3000	32	Acc
GSM8K	1319	128	EM
HumanEval	164	128	Pass@1

Table 4: Statistics of datasets.

## A Evaluated Benchmarks

We summarize the six benchmarks used in this work in Table 4. AlpacaEval 2 (Li et al., 2023) and Arena-Hard (Li et al., 2024b) are general instruction-following benchmarks. AlpacaEval consists of 805 questions, and Arena-Hard incorporating 500 well-defined technical problem-solving queries. For AlpacaEval 2, we report the length-controlled win rate (LC). For Arena-Hard, we report the win rate (WR) against the baseline model.

Since the original MMLU (Hendrycks et al., 2020) benchmark is huge and contain numerous ground truth errors (Wang et al., 2024; Gema et al., 2024), we use MMLU-Redux (Gema et al., 2024) which is a subset of 3000 manually re-annotated questions across 30 MMLU subjects. We also include GSM8K (Cobbe et al., 2021), and HumanEval (Chen et al., 2021), two popular benchmarks for evaluating the math and code generation abilities of LLMs.

## B Evaluation Setup

We use official evaluation scripts for AlpacaEval 2, Arena-Hard, and MixEval. For MMLU-Redux, instead of using the next token probability of the choice letters, we encourage the model to generate the answer in the form of natural language sentence. For GSM8K and HumanEval, we use Open-Instruct framework (Wang et al., 2023) to evaluate the models. For non-deterministic sampling, to encourage the models to generate more diverse completions, we use nucleus sampling and set the generation temperature to 1.0 and top-p to 1.0. We sample 16 completions for AlpacaEval 2, Arena-Hard, and MixEval, 32 completions for MMLU-Redux, 128 for GSM8K and HumanEval.