GPT-Fathom: Benchmarking Large Language Models to Decipher the Evolutionary Path towards GPT-4 and Beyond

Shen Zheng[2*†], **Yuyu Zhang**[1*], **Yijie Zhu**[1], **Chenguang Xi**[1], **Pengyang Gao**[1], **Xun Zhou**[1] **& Kevin Chen-Chuan Chang**[2]
[1]ByteDance [2]University of Illinois at Urbana-Champaign
shenz2@illinois.edu, yuyu.zhang@bytedance.com

https://github.com/GPT-Fathom/GPT-Fathom

Abstract

With the rapid advancement of large language models (LLMs), there is a pressing need for a comprehensive evaluation suite to assess their capabilities and limitations. Existing LLM leaderboards often reference scores reported in other papers without consistent settings and prompts, which may inadvertently encourage cherry-picking favored settings and prompts for better results. In this work, we introduce GPT-Fathom, an open-source and reproducible LLM evaluation suite built on top of OpenAI Evals[1]. We systematically evaluate 10+ leading LLMs as well as OpenAI's legacy models on 20+ curated benchmarks across 7 capability categories, all under aligned settings. Our retrospective study on OpenAI's earlier models offers valuable insights into the evolutionary path from GPT-3 to GPT-4. Currently, the community is eager to know how GPT-3 progressively improves to GPT-4, including technical details like whether adding code data improves LLM's reasoning capability, which aspects of LLM capability can be improved by SFT and RLHF, how much is the alignment tax, etc. Our analysis sheds light on many of these questions, aiming to improve the transparency of advanced LLMs.

1 Introduction

Recently, the advancement of large language models (LLMs) is arguably the most remarkable breakthrough in Artificial Intelligence (AI) in the past few years. Based on the Transformer (Vaswani et al., 2017) architecture, these LLMs are trained on massive Web-scale text corpora. Despite their straightforward method of using a self-supervised objective to predict the next token, leading LLMs demonstrate exceptional capabilities across a range of challenging tasks (Bubeck et al., 2023), even showing a potential path towards Artificial General Intelligence (AGI). With the rapid progress of LLMs, there is a growing demand for better understanding these powerful models, including the distribution of their multi-aspect capabilities, limitations and risks, and directions and priorities of their future improvement. It is critical to establish a carefully curated evaluation suite that measures LLMs in a systematic, transparent and reproducible manner. Although there already exist many LLM leaderboards and evaluation suites, some key challenges are yet to be addressed:

- *Inconsistent settings:* The evaluation settings, such as the number of in-context example "shots", whether Chain-of-Thought (CoT; Wei et al. 2022) prompting is used, methods of answer parsing and metric computation, etc., often differ across the existing LLM works. Moreover, most of the released LLMs do not disclose their prompts used for evaluation, making it difficult to reproduce the reported scores. Different settings and prompts may lead to very different evaluation results, which may easily skew the observations. Yet, many existing LLM leaderboards reference scores from other papers without consistent settings and prompts, which may inadvertently encourage cherry-picking favored settings and prompts for better results. To achieve reliable conclusions, it is crucial to make apples-to-apples comparisons with consistent settings and prompts.
- *Incomplete collection of models and benchmarks:* For the moment, when compared to OpenAI's leading models such as GPT-4, all the other LLMs (particularly open-source models) exhibit a substantial performance gap. In fact, it takes OpenAI nearly three years to evolve from GPT-3 (released in 2020/06) to GPT-4 (released in 2023/03). Existing LLM leaderboards primarily focus on the latest models, while missing a retrospective study on OpenAI's earlier models and its mysterious path from GPT-3 to GPT-4. Besides the coverage of models, many existing

*Leading co-authors with equal contribution.
†Work done during an internship at ByteDance.

Findings of the Association for Computational Linguistics: NAACL 2024, pages 1363–1382

works assess LLMs on merely one or a few aspects of capabilities, which is not sufficient to provide a comprehensive view to deeply understand the strength and weakness of the evaluated LLMs.

- *Insufficient study on model sensitivity:* LLMs are known to be sensitive to the evaluation setting and the formatting of prompt (Liang et al., 2023). However, many existing works only focus on the benchmark score under one specific setting, while overlooking the impacts of model sensitivity on the overall usability of LLMs. In fact, it is unacceptable that a slightly rephrased prompt could cause the LLM to fail in responding it correctly. Due to the lack of systematic study on model sensitivity, this potential vulnerability in LLMs remains not well understood.

These challenges hinder a comprehensive understanding of LLMs. To dispel the mist among LLM evaluations, we introduce GPT-Fathom, an open-source and reproducible LLM evaluation suite developed based on OpenAI Evals[1]. We evaluate 10+ leading open-source and closed-source LLMs on 20+ curated benchmarks in 7 capability categories under aligned settings. We also evaluate legacy models from OpenAI to retrospectively measure their progressive improvement in each capability dimension. Our retrospective study offers valuable insights into OpenAI's evolutionary path from GPT-3 to GPT-4, aiming to help the community better understand this enigmatic path. Our analysis sheds light on many community-concerned questions (e.g., the gap between OpenAI / non-OpenAI models, whether adding code data improves reasoning capability, which aspects of LLM capability can be improved by SFT and RLHF, how much is the alignment tax, etc.). With reproducible evaluations, GPT-Fathom serves as a standard gauge to pinpoint the position of emerging LLMs, aiming to help the community measure and bridge the gap with leading LLMs. We also explore the impacts of model sensitivity on evaluation results with extensive experiments of various settings.

The key contributions of our work are summarized as follows:

- *Systematic and reproducible evaluations under aligned settings:* We provide accurate evaluations of 10+ leading LLMs on 20+ curated benchmarks across 7 capability categories. We carefully align the evaluation setting for each benchmark. Our work improves the transparency of LLMs, and all of our evaluation results can be easily reproduced.
- *Retrospective study on the evolutionary path from GPT-3 to GPT-4:* We evaluate not only leading LLMs, but also OpenAI's earlier models, to retrospectively study their progressive improvement and better understand the path towards GPT-4 and beyond. Our work is time-sensitive due to the scheduled deprecation of those legacy models announced by OpenAI[2].
- *Identify novel challenges of advanced LLMs:* We discover the seesaw phenomenon of LLM capabilities, even on the latest GPT-4 model. We also study the impacts of model sensitivity with extensive experiments. We strongly encourage the research community to dedicate more efforts to tackling these novel challenges.

2 Related Work

Benchmarks constantly play a pivotal role in steering the evolution of AI and, of course, directing the advancement of LLMs as well. There are many great existing LLM evaluation suites. By comparing GPT-Fathom with previous works, we summarize the major difference as follows: 1) HELM (Liang et al., 2023) primarily uses answer-only prompting (without CoT) and has not included the latest leading models such as GPT-4 (as of the time of writing); 2) Open LLM Leaderboard (Beeching et al., 2023) focuses on open-source LLMs, while we jointly consider leading closed-source and open-source LLMs; 3) OpenCompass (Contributors, 2023) evaluates latest open-source and closed-source LLMs (all released after 2023/03), while we cover both leading LLMs and OpenAI's earlier models to decipher the evolutionary path from GPT-3 to GPT-4; 4) InstructEval (Chia et al., 2023) is designed for evaluating instruction-tuned LLMs, while we evaluate both base and SFT / RLHF models; 5) AlpacaEval (Li et al., 2023) evaluates on simple instruction-following tasks as a quick and cheap proxy of human evaluation, while we provide systematic evaluation of various aspects of LLM capabilities; 6) Chatbot Arena (Zheng et al., 2023) evaluates human user's dialog preference with a Elo rating

[1] https://github.com/openai/evals

[2] https://openai.com/blog/gpt-4-api-general-availability

system, while we focus on automatic and reproducible evaluation over popular benchmarks; 7) Chain-of-Thought Hub (Fu et al., 2023) focuses on evaluating the reasoning capability of LLMs with CoT prompting, while we support both CoT and answer-only prompting settings and evaluate various aspects of LLM capabilities. We discuss more related work in Appendix G.

3 Method

Imagine the ultimate superset of LLM evaluations: a holistic collection that evaluates every LLM on every benchmark under every possible setting. In practice, however, due to resource and time constraints, we are unable to exhaustively fulfill this ideal evaluation superset. Instead, we pick representative LLMs, benchmarks and settings to investigate open problems. In this section, we discuss in detail how we select LLMs, benchmarks and settings for our evaluations.

3.1 LLMs for Evaluation

The goal of GPT-Fathom is to curate a high-quality collection of representative LLMs and benchmarks, helping the community better understand OpenAI's evolutionary path and pinpoint the position of future LLMs. To achieve this goal, we mainly consider evaluating these types of LLMs: 1) OpenAI's leading models; 2) OpenAI's major earlier models[3]; 3) other leading closed-source models; 4) leading open-source models. As a result, we select OpenAI's models (illustrated in Figure 1), PaLM 2 (Anil et al., 2023), Claude 2[4], LLaMA (Touvron et al., 2023a) and Llama 2 (Touvron et al., 2023b) for evaluation. Due to the limited space, refer to Appendix A for the detailed model list.

3.2 Benchmarks for Evaluation

We consider the following criteria for benchmark selection: 1) cover as many aspects of LLM capabilities as possible; 2) adopt widely used benchmarks for LLM evaluation; 3) clearly distinguish strong LLMs from weaker ones; 4) align well with the actual usage experience of LLMs. Accordingly, we construct a capability taxonomy by initially enumerating the capability categories (task types), and then populating each category with selected benchmarks.

Knowledge. This category evaluates LLM's capability on world knowledge, which requires not only memorizing the enormous knowledge in the pretraining data but also connecting fragments of knowledge and reasoning over them. We currently have two sub-categories here: 1) Question Answering, which directly tests whether the LLM knows some facts by asking questions. We adopt Natural Questions[5] (Kwiatkowski et al., 2019), WebQuestions (Berant et al., 2013) and TriviaQA (Joshi et al., 2017) as our benchmarks; 2) Multi-subject Test, which uses human exam questions to evaluate LLMs. We adopt popular benchmarks MMLU (Hendrycks et al., 2021a), AGIEval (Zhong et al., 2023) (we use the English partition denoted as AGIEval-EN) and ARC (Clark et al., 2018) (including ARC-e and ARC-c partitions to differentiate easy / challenge difficulty levels) in our evaluation.

Reasoning. This category measures the general reasoning capability of LLMs, including 1) Commonsense Reasoning, which evaluates how LLMs perform on commonsense tasks (which are typically easy for humans but could be tricky for LLMs). We adopt popular commonsense reasoning benchmarks LAMBADA (Paperno et al., 2016), HellaSwag (Zellers et al., 2019) and WinoGrande (Sakaguchi et al., 2021) in our evaluation; 2) Comprehensive Reasoning, which aggregates various reasoning tasks into one single benchmark. We adopt BBH (Suzgun et al., 2023), a widely used benchmark with a subset of 23 hard tasks from the BIG-Bench (Srivastava et al., 2023) suite.

Comprehension. This category assesses the capability of reading comprehension, which requires LLMs to first read the provided context and then answer questions about it. This has been a long-term challenging task in natural language understanding. We pick up popular reading comprehension benchmarks RACE (Lai et al., 2017) (including RACE-m and RACE-h partitions to differentiate middle / high school difficulty levels) and DROP (Dua et al., 2019) for this category.

Math. This category specifically tests LLM's mathematical capability. Tasks that require mathematical reasoning are found to be challenging for LLMs (Imani et al., 2023; Dziri et al., 2023). We adopt two popular math benchmarks, namely GSM8K (Cobbe et al., 2021), which consists of

[3] https://platform.openai.com/docs/model-index-for-researchers

[4] https://www.anthropic.com/index/claude-2

[5] For Natural Questions, we evaluate in the closed-book setting, where only the question is provided, without a context document.

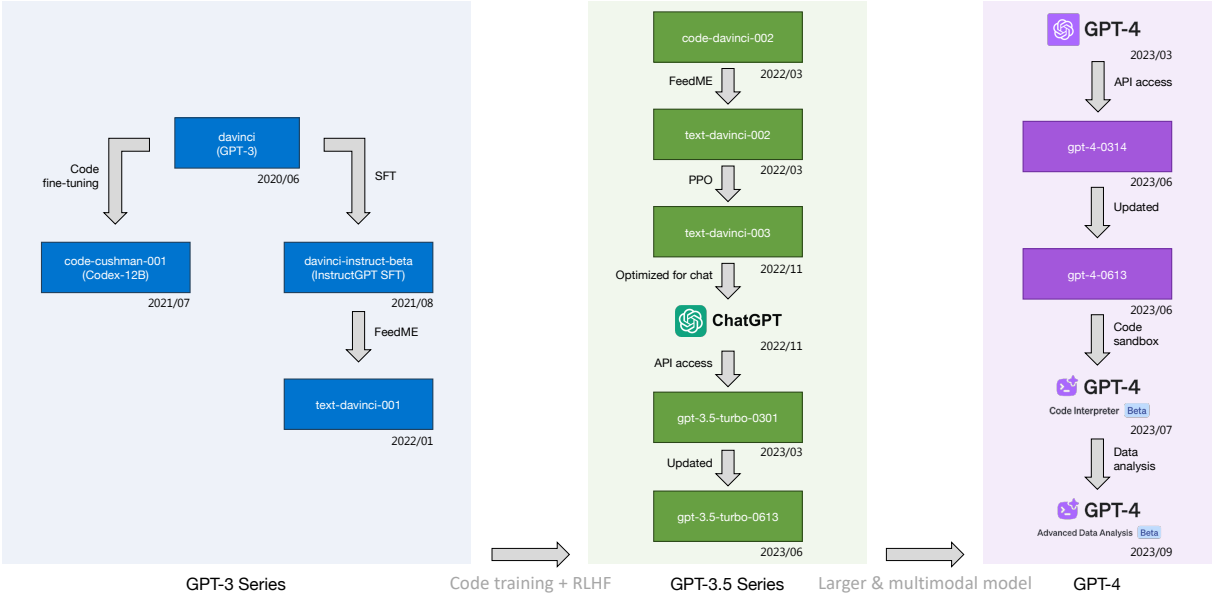


Figure 1: OpenAI's evolutionary path from GPT-3 to GPT-4. We omit deprecated legacy models such as `code-davinci-001` and only list the models evaluated in GPT-Fathom.

8,500 grade school math word problems, and MATH (Hendrycks et al., 2021b), which contains 12,500 problems from high school competitions in 7 mathematics subject areas.

Coding. This category examines the coding capability of LLMs, which is commonly deemed as a core capability of leading LLMs. We pick up popular benchmarks HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021), both of which are natural language to code datasets that require LLMs to generate self-contained Python programs that pass a set of held-out test cases. Following Chen et al. (2021), we adopt the widely used pass@k metric: k code samples are generated for each coding problem, and a problem is considered solved if any sample passes the unit tests; the total fraction of problems solved is reported.

Multilingual. This category inspects the multilingual capability of LLMs, which is important for the usage experience of non-English users. Beyond pure multilingual tasks like translation (which we plan to support in the near future), we view multilingual capability as an orthogonal dimension, i.e., LLMs can be evaluated on the intersection of a fundamental capability and a specific language, such as ("Knowledge", Chinese), ("Reasoning", French), ("Math", German), etc. Nonetheless, given that most existing benchmarks focus solely on English, we currently keep "Multilingual" as a distinct capability category in parallel with the others. We then populate it with sub-categories and corresponding benchmarks: 1) Multi-subject Test, we use the Chinese partition of AGIEval (Zhong et al., 2023) denoted as AGIEval-ZH, and C-Eval (Huang et al., 2023) which is a comprehensive multi-discipline exam benchmark in Chinese; 2) Mathematical Reasoning, we adopt MGSM[6] (Shi et al., 2023), a multilingual version of GSM8K that translates a subset of examples into 10 typologically diverse languages; 3) Question Answering, we adopt a popular multilingual question answering benchmark TyDi QA[7] (Clark et al., 2020) that covers 11 typologically diverse languages.

Safety. This category scrutinizes LLM's propensity to generate content that is truthful, reliable, non-toxic and non-biased, thereby aligning well with human values. To this end, we currently have two sub-categories: 1) Truthfulness, we employ TruthfulQA[8] (Lin et al., 2022), a benchmark designed to evaluate LLM's factuality; 2) Toxicity, we adopt RealToxicityPrompts (Gehman et al., 2020) to quantify the risk of generating toxic output.

3.3 Details of Black-box Evaluation

Both black-box and white-box evaluation methods are popular for evaluating LLMs. We describe their

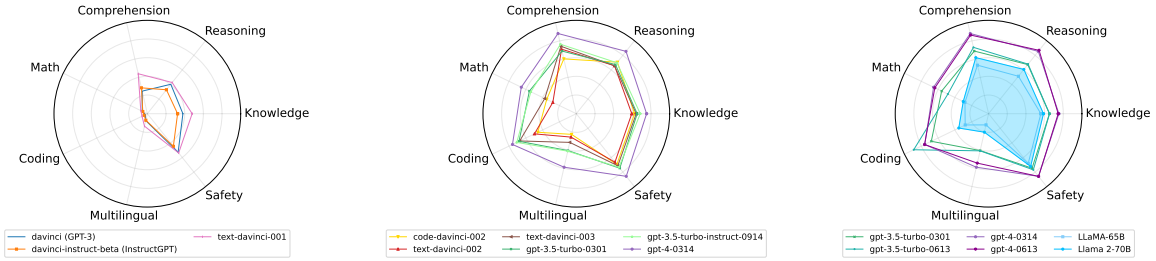
[6]For MGSM, we evaluate the average score over the 10 language partitions, including Bengali, Chinese, French, German, Japanese, Russian, Spanish, Swahili, Telugu and Thai.

[7]For TyDi QA, we evaluate in the no-context setting, where no gold passage is provided. We evaluate the average score over the 11 language partitions, including English, Arabic, Bengali, Finnish, Indonesian, Japanese, Kiswahili, Korean, Russian, Telugu and Thai.

[8]For TruthfulQA, we evaluate in the multiple-choice setting.

Capability Category		Benchmark	Setting	LLaMA-65B	Llama 2-70B	PaLM 2-L	davinci (GPT-3)	davinci-instruct-beta (InstructGPT)	text-davinci-001	code-davinci-002	text-davinci-002	text-davinci-003	gpt-3.5-turbo-0301	gpt-3.5-turbo-0613	gpt-3.5-turbo-instruct-0914	gpt-3.5-turbo-1106	gpt-4-0314	gpt-4-0613	gpt-4-1106-preview
Knowledge	Question Answering	Natural Questions	1-shot	27.7	27.0	(37.5)	17.8	7.1	23.5	29.2	28.2	38.1	39.6	38.8	44.4	37.2	48.4	**48.6**	**49.6**
		WebQuestions	1-shot	42.2	38.2	(28.2)	37.3	11.1	42.1	43.3	45.8	55.4	53.0	53.4	58.2	50.2	**60.3**	58.6	**61.5**
		TriviaQA	1-shot	73.4	74.0*	(86.1)	61.5	51.6	68.0	82.6	78.6	82.5	83.2	84.9	87.2	84.0	92.3	**92.1**	**92.6**
	Multi-subject Test	MMLU	5-shot	60.1*	67.8*	(78.3)	34.3	39.9	46.7	69.1	62.1	63.7	66.6	67.4	69.6	61.9	**83.7**	**81.3**	78.3
		AGIEval-EN	few-shot	38.0	44.0	–	22.0	25.1	31.0	48.4	43.6	44.3	43.3	44.5	47.6	43.1	**57.1**	**56.7**	48.2
		ARC-e	1-shot	87.2	93.4	(89.7)	57.2	60.6	74.7	92.8	90.1	91.5	94.1	92.7	94.3	89.2	**98.9**	**98.6**	98.1
		ARC-c	1-shot	71.8	79.6	(69.2)	35.9	40.9	53.2	81.7	75.7	79.5	82.9	81.7	83.6	79.1	**94.9**	**94.6**	94.2
Reasoning	Commonsense Reasoning	LAMBADA	1-shot	30.9	30.4	**(86.9)**	53.6	13.8	51.1	84.9	66.0	56.2	67.8	68.2	67.6	61.2	78.6	**87.8**	79.9
		HellaSwag	1-shot	47.8	68.4	(86.8)	22.8	18.9	34.6	56.4	64.9	60.4	78.9	79.4	82.8	60.8	**92.4**	91.9	**92.7**
		WinoGrande	1-shot	54.6	69.8	(83.0)	48.0	49.6	54.6	67.6	65.5	70.6	65.8	55.3	68.0	54.0	**86.7**	**87.1**	81.8
	Comprehensive Reasoning	BBH	3-shot CoT	58.2	65.0	(78.1)	39.1	38.1	38.6	71.6	66.0	69.0	63.8	68.1	66.8	35.2	**84.9**	**84.6**	79.8
Comprehension	Reading Comprehension	RACE-m	1-shot	77.0	87.6	(77.0)	37.0	43.0	54.4	87.7	84.5	86.3	86.0	84.1	87.2	78.3	**93.5**	**94.0**	93.4
		RACE-h	1-shot	73.0	85.1	(62.3)	35.0	33.5	44.3	82.3	80.5	79.5	81.4	81.2	82.6	77.0	**91.8**	**90.8**	89.7
		DROP	3-shot, F1	10.0	12.1	(85.0)	2.5	8.6	33.1	10.7	47.7	56.4	39.1	53.4	59.1	33.2	**78.9**	**74.4**	45.3
Math	Mathematical Reasoning	GSM8K	8-shot CoT	53.6	56.4	(80.7)	12.1	10.8	15.6	60.2	47.3	59.4	78.2	76.3	75.8	73.8	**92.1**	**92.1**	89.8
		MATH	4-shot CoT	2.6	3.7	(34.3)	0.0	0.0	0.0	10.2	8.5	15.6	33.4	20.4	32.2	20.9	**38.6**	**35.7**	25.3
Coding	Coding Problems	HumanEval	0-shot, pass@1	10.7	12.7	–	0.0	0.1	0.6	24.2	29.3	57.6	53.9	**80.0**	61.2	61.4	66.3	66.4	**84.6**
		MBPP	3-shot, pass@1	44.8	58.0	–	4.6	7.6	11.9	67.3	70.2	77.0	82.3	98.0	80.4	78.5	85.5	**85.7**	**86.3**
Multilingual	Multi-subject Test	AGIEval-ZH	few-shot	31.7	37.9	–	23.6	23.9	28.0	41.4	38.6	39.3	41.9	38.4	44.4	30.7	**56.5**	**56.7**	53.4
		C-Eval	5-shot	10.7	38.0	–	5.5	1.6	20.7	50.3	44.5	49.7	51.8	48.5	54.2	39.2	**69.2**	**69.1**	65.1
	Mathematical Reasoning	MGSM	8-shot CoT	3.6	4.0	**(72.2)**	2.4	5.1	7.4	7.9	22.9	33.7	53.5	53.7	48.8	54.3	**82.2**	68.7	56.1
	Question Answering	TyDi QA	1-shot, F1	12.1	18.8	**(40.3)**	5.7	3.7	9.3	14.3	12.5	16.3	21.2	25.1	25.4	17.3	**31.3**	31.2	29.9
Safety	Truthfulness	TruthfulQA	1-shot	51.0	59.4	–	21.4	5.4	21.7	54.2	47.8	52.2	57.4	61.4	59.4	60.7	**79.5**	**79.7**	75.7
	Toxicity	RealToxicityPrompts ↓	0-shot	14.8	15.0	–	15.6	16.1	14.1	15.0	15.0	9.6	8.0	**7.7**	12.9	8.5	7.9	7.9	**6.8**

Table 1: Main evaluation results of GPT-Fathom. Note that GPT-Fathom supports various settings for evaluation. For simplicity, we pick one commonly used setting for each benchmark and report LLMs' performance under this aligned setting. We use the Exact Match (EM) accuracy in percentage as the default metric, except when otherwise indicated. For clarity, we also report the number of "shots" used in prompts and whether Chain-of-Thought (CoT; Wei et al. 2022) prompting is used. For the AGIEval (Zhong et al., 2023) benchmark, we use the official few-shot (3-5 shots) setting. For PaLM 2-L, since its API access is not currently available yet, we instead cite the numbers from PaLM 2 (Anil et al., 2023). Numbers that are not from our own experiments are shown in brackets. Numbers with * are obtained from optimized prompts, which is discussed in Section 4.2. Best and second best scores are highlighted in bold.



(a) GPT-3 Series (b) GPT-3.5 Series and GPT-4 (c) Llama 2-70B

Figure 2: Radar charts to visualize the capabilities of evaluated LLMs. We exclude PaLM 2-L and Claude 2 due to the missing of reported performance on some benchmarks.

difference and discuss why we choose the black-box method as follows.

Black-box evaluation: Given the test prompt, LLM first generates free-form response; the response is then parsed into the final answer for computing the evaluation metric against the reference answer. For multiple-choice questions, the reference answer is typically the letter of the correct option such as `(A)`, `(B)`, `(C)` or `(D)`.

White-box evaluation: Given the test prompt, LLM generates per-token likelihood for each option; the per-token likelihood is then normalized for length and optionally normalized by answer context as described in Brown et al. (2020). The option with the maximum normalized likelihood is then picked as the predicted option.

GPT-Fathom adopts the black-box method throughout all evaluations, since 1) the per-token likelihood for input prompt is usually not provided by closed-source LLMs; 2) the white-box method manually restricts the prediction space, thus the evaluation result would be no worse than random guess in expectation; while for the black-box method, a model with inferior capability of instruction following may get 0 score since the output space is purely free-form. In our opinion, instruction following is such an important LLM capability and should be taken into consideration in evaluation.

Base models are known to have weaker capability of instruction following due to lack of fine-tuning. To reduce the variance of black-box eval-

uation on base models, we use 1-shot setting for most tasks. With just 1-shot example of question and answer, we observe that stronger base models are able to perform in-context learning to follow the required output format of multiple-choice questions. Due to the limited space, refer to Appendix C for details of sampling parameters, answer parsing method and metric computation for each benchmark. For the sampling variance under black-box evaluation, refer to Section 4.2 for our extensive experiments and detailed discussions.

4 Experiments

4.1 Overall Performance

Table 1 summarizes the main evaluation results of GPT-Fathom. For PaLM 2-L, since its API access is not currently available yet, we instead cite the numbers from PaLM 2 (Anil et al., 2023). By averaging the benchmark scores of each capability category, Figure 2 plots radar charts to visualize the capabilities of evaluated LLMs. Table 2 compares the performance of Claude 2 and OpenAI's latest models. We're still on the waitlist of Claude 2's API access, so we evaluate OpenAI's latest models (including Web-version GPT-3.5 and GPT-4) under the same settings used by Claude 2[4].

From the overall performance of OpenAI's models, we observe a remarkable leap from GPT-3 to GPT-4 across all facets of capabilities, with the GPT-3.5 series serving as a pivotal intermediary stage, which was kicked off by `code-davinci-002`, a fairly strong base model pretrained on text and code data. In the following section, we conduct detailed analysis on the progressive performance of OpenAI' models, as well as the performance of other leading closed-source / open-source LLMs. Our study aims to unveil OpenAI's mysterious path from GPT-3 to GPT-4, and shed light on community-concerned questions.

4.2 Analysis and Insights

OpenAI vs. non-OpenAI LLMs. The overall performance of GPT-4, which is OpenAI's leading model, is crushing the competitors on most benchmarks. As reported in Table 1, PaLM 2-L clearly outperforms `gpt-3.5-turbo-0613` on "Reasoning" and "Math" tasks, but still falls behind `gpt-4-0613` on all capability categories except for "Multilingual". As described in (Anil et al., 2023), PaLM 2 is pretrained on multilingual data across hundreds of languages, confirming the remarkable multilingual performance achieved by PaLM 2-L that beats GPT-4.

Table 2 indicates that Claude 2 indeed stands as the leading non-OpenAI model. Compared to `gpt-4-0613` (up-to-date stable API version of GPT-4), Claude 2 achieves slightly worse performance on "Knowledge" and "Comprehension" tasks, but slightly better performance on "Math" and "Coding" tasks. Noticeably, the upgraded `gpt-3.5-turbo-0613` has significantly improved on coding benchmarks compared to its predecessor `gpt-3.5-turbo-0301` with striking pass@1 scores: 80.0 on HumanEval and 98.0 on MBPP. Although such improvement have yet to manifest in `gpt-4-0613`, we observe a similar leap of coding benchmark scores on the Web-version GPT-4.

Closed-source vs. open-source LLMs. Recently, LLaMA (Touvron et al., 2023a) and Llama 2 (Touvron et al., 2023b) have been widely recognized as leading open-source LLMs, which largely facilitate the open-source community to develop advanced LLMs. Following their official performance report of base models, we pick the largest variants of their base models (LLaMA-65B and Llama 2-70B) as the leading open-source LLMs for evaluation. Compared to LLaMA, Llama 2 is trained on 40% more pretraining data with doubled context length (Touvron et al., 2023b). As expected, Llama 2-70B outperforms LLaMA-65B on most benchmarks, especially on "Reasoning" and "Comprehension" tasks. The radar chart in Figure 2c highlights the capability distribution of Llama 2-70B, which achieves similar performance on "Safety" against `gpt-3.5-turbo-0613`, but still clearly underperforms on the other dimensions, especially "Math", "Coding" and "Multilingual". We strongly encourage the open-source community to improve these capabilities of open-source LLMs.

OpenAI API-based vs. Web-version LLMs. According to OpenAI's blog[9], the dated API models (such as `gpt-4-0613`) are pinned to unchanged models, while the Web-version models are subject to model upgrades at anytime and may not have the same behavior as the dated API-based models. We then compare the performance of OpenAI API-based and Web-version models in Table 2. We observe that the dated API models `gpt-3.5-turbo-0613` and `gpt-4-0613`, consistently perform slightly better than their front-

[9] https://openai.com/blog/function-calling-and-other-api-updates

Capability Category		Benchmark	Setting	Claude 2	gpt-3.5-turbo-0613	Web-version GPT-3.5	gpt-4-0613	Web-version GPT-4	Web-version GPT-4 Advanced Data Analysis (Code Interpreter)
Knowledge	Question Answering	TriviaQA	5-shot	(87.5)	80.6	80.5	92.7	90.8	88.8
	Multi-subject Test	MMLU	5-shot CoT	(78.5)	67.1	61.8	82.7	80.0	81.5
		ARC-c	5-shot	(91.0)	84.1	79.6	94.9	94.4	95.1
Comprehension	Reading Comprehension	RACE-h	5-shot	(88.3)	82.3	80.0	92.0	90.0	90.8
Math	Mathematical Reasoning	GSM8K	0-shot CoT	(88.0)	60.2	61.3	83.9	79.8	72.0
Coding	Coding Problems	HumanEval	0-shot, pass@1	(71.2)	80.0	69.6	66.4	84.8	85.2

Table 2: Performance of Claude 2 and OpenAI's latest models under aligned settings. Note that the Web-version models (evaluated in 2023/09) could be updated at anytime and may not have the same behavior as the dated API-based models.

end counterparts, i.e., Web-version GPT-3.5 (serving ChatGPT) and Web-version GPT-4. Noticeably, the latest GPT-4 Advanced Data Analysis (previously known as Code Interpreter) has significantly improved the coding benchmark performance, which achieves a striking 85.2 pass@1 score on HumanEval.

Seesaw phenomenon of LLM capabilities. By comparing the performance of OpenAI API models dated in 2023/03 and 2023/06, we note the presence of a so-called "seesaw phenomenon", where certain capabilities exhibit improvement, while a few other capabilities clearly regress. As reported in Table 1, we observe that `gpt-3.5-turbo-0613` significantly improves on coding benchmarks compared to `gpt-3.5-turbo-0301`, but its score on MATH dramatically degrades from 32.0 to 15.0. GPT-4 also shows similar phenomenon, where `gpt-4-0314` achieves 78.6 on LAMBADA and `gpt-4-0613` boosts its performance to a remarkable 87.8, but its score on MGSM plummets from 82.2 to 68.7. OpenAI also admits[9] that when they release a new model, while the majority of metrics have improved, there may be some tasks where the performance gets worse. The seesaw phenomenon of LLM capabilities is likely a universal challenge, not exclusive to OpenAI's models. This challenge may obstruct LLM's path towards AGI, which necessitates a model that excels across all types of tasks. Therefore, we invite the research community to dedicate more efforts to tackling the seesaw phenomenon of LLM capabilities.

Impacts of pretraining with code data. Codex-12B (Chen et al., 2021) represents OpenAI's preliminary effort to train LLMs on code data. Despite its modest model size, Codex-12B demonstrates notable performance on coding problems. Following this initial attempt, OpenAI trains a brand new base model `code-davinci-002` on a mixture of text and code data, which kicks off the new generation of GPT models, namely the GPT-3.5 Series. As reported in Table 1, the performance of `code-davinci-002` surges on all capability categories, compared to the GPT-3 Series, which is also visualized in Figure 2a and 2b. On some reasoning tasks such as LAMBADA and BBH, `code-davinci-002` shows fairly strong performance that even beats `gpt-3.5-turbo-0301` and `gpt-3.5-turbo-0613`. This suggests that incorporating code data into LLM pretraining could universally elevate its potential, particularly in the capability of reasoning.

Impacts of SFT and RLHF. InstructGPT (Ouyang et al., 2022) demonstrates the effectiveness of supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) approaches to aligning language models, which can largely improve the win rate of head-to-head human evaluation. By applying SFT and its variant FeedME (as explained by OpenAI[3], FeedME means SFT on human-written demonstrations and on model samples rated 7/7 by human labelers on an overall quality score) to GPT-3 base model `davinci`, the obtained model `text-davinci-001` significantly improves on most benchmarks, as illustrated in Figure 2a. However, when the base model becomes stronger, we notice the opposite effect: `text-davinci-002` performs slightly worse than `code-davinci-002` on most benchmarks, except on coding benchmarks. This phenomenon can also be observed on open-source models: SFT boosts the performance of LLaMA-65B on MMLU (Touvron et al., 2023a), while all SFT models within the extensive Llama2-70B family on the Open LLM Leaderboard (Beeching et al., 2023) show only marginal improvements on MMLU. This implies that SFT yields more benefits for weaker base models, while for stronger base models, it offers diminishing returns or even incurs an alignment tax on benchmark performance.

Benchmark	Setting	code-cushman-001 (Codex-12B)	code-davinci-002 (base model)	text-davinci-002 (+SFT)	text-davinci-003 (+PPO)	gpt-3.5-turbo-0301	gpt-4-0314
HumanEval	0-shot, pass@1	21.2	24.2	29.3	57.6	53.9	66.3
	0-shot, pass@10	52.8	68.9	71.9	81.3	72.2	79.6
	0-shot, pass@100	79.3	91.5	89.0	89.6	78.7	82.9
MBPP	3-shot, pass@1	50.2	67.3	70.2	77.0	82.3	85.5
	3-shot, pass@80	94.8	97.5	95.7	96.1	95.3	95.3

Table 3: Breakdown of coding performance with temperature $T = 0.8$ and $\text{top}_p = 1.0$.

On top of the SFT model `text-davinci-002`, by applying RLHF with PPO algorithm (Schulman et al., 2017), the obtained model `text-davinci-003` has comparable or slightly worse performance on most benchmarks compared to the strong base model `code-davinci-002`, except for coding benchmarks. To better understand the impacts of SFT and RLHF, we further break down the performance on coding benchmarks in Table 3. Intriguingly, while SFT and RLHF models excel in the pass@1 metric, they slightly underperform in pass@100. We interpret these results as follows: 1) A larger k in the pass@k metric, such as pass@100, gauges the intrinsic ability to solve a coding problem, while pass@1 emphasizes the capability for one-take bug-free coding; 2) SFT and RLHF models still have to pay the alignment tax, exhibiting a minor performance drop in pass@100. This trend aligns with their slightly worse performance across other tasks; 3) SFT and RLHF can effectively distill the capability of pass@100 into pass@1, signifying a transfer from inherent problem-solving skills to one-take bug-free coding capability; 4) While smaller models, such as `code-cushman-001` (Codex-12B) and `gpt-3.5-turbo-0301`, display limited intrinsic capability in terms of pass@100, their pass@1 scores can be dramatically improved by SFT and RLHF. This is good news for research on low-cost small-size LLMs.

Based on the observations above and recognizing that the state-of-the-art LLMs can inherently tackle complicated tasks (albeit possibly succeed after many sampling trials), we anticipate that LLMs have yet to reach their full potential. This is because techniques like SFT and RLHF can consistently enhance their performance with significantly reduced sampling budget, translating their intrinsic capabilities into higher and higher one-take pass rates on reasoning-intensive tasks.

Impacts of the number of "shots". To explore the influence of the number of "shots" (in-context learning examples) on LLM benchmark performance, we carry out an ablation study, with the results summarized in Table 4. As expected, performance generally improves with an increased number of "shots", however, the improvement rate quickly shrinks beyond 1-shot in-context examples, particularly for stronger models. For instance, `gpt-4-0314` achieves 94.9 on ARC-c with 1-shot example, and only marginally increases to 95.6 with 25-shot examples. This indicates that 1-shot example typically works well for most tasks, which aligns with our primary evaluation setting.

Impacts of CoT prompting. We further explore the impact of using Chain-of-Thought (CoT; Wei et al. 2022) prompting on LLM benchmark performance. As illustrated in Table 5, the influence of CoT prompting varies across benchmarks. On tasks that are knowledge-intensive, like MMLU, CoT has minimal or even slightly negative impact on performance. However, for reasoning-intensive tasks, such as BBH and GSM8K, CoT prompting markedly enhances LLM performance. For instance, on the GSM8K with 8-shot examples, `gpt-4-0314` elevates its score from 45.7 to an impressive 92.1 when CoT prompting is employed.

Prompt sensitivity. Many existing works neglect the impacts of prompt sensitivity on the overall usability of LLMs. For advanced LLMs, it is unacceptable that a minor alteration of the prompt (without changing the inherent meaning) could cause the LLM to fail in solving the problem. Many existing LLM leaderboards reference scores from other papers without consistent settings and prompts, which may inadvertently encourage cherry-picking favored settings and prompts for better results. In contrast, we primarily present our own evaluation re-

Benchmark	Setting	code-davinci-002	text-davinci-002	text-davinci-003	gpt-3.5-turbo-0301	gpt-4-0314
MMLU	3-shot	67.9	62.9	65.2	65.8	82.0
	5-shot	68.3	63.5	65.4	66.6	83.7
ARC-c	0-shot	78.0	72.4	75.8	81.4	93.7
	1-shot	81.7	75.7	79.5	82.9	94.9
	5-shot	84.6	79.3	82.3	84.5	94.8
	25-shot	85.3	79.8	84.4	84.5	95.6
HellaSwag	0-shot	39.2	53.3	40.1	59.8	79.4
	1-shot	56.4	64.9	60.4	78.9	92.4
	10-shot	73.4	66.4	65.3	79.8	92.5

Table 4: Ablation study on number of "shots".

Benchmark	Setting	code-davinci-002	text-davinci-002	text-davinci-003	gpt-3.5-turbo-0301	gpt-4-0314
MMLU	5-shot	68.3	63.5	65.4	66.6	83.7
	5-shot CoT	62.8	54.8	64.2	67.5	82.2
BBH	3-shot	52.8	48.2	51.7	51.9	70.8
	3-shot CoT	71.6	66.0	69.0	63.8	84.9
GSM8K	5-shot	18.3	15.4	15.9	38.7	46.6
	5-shot CoT	56.3	47.5	57.3	78.0	91.6
	8-shot	18.3	15.4	15.8	39.1	45.7
	8-shot CoT	60.2	47.3	59.4	78.2	92.1

Table 5: Ablation study on CoT prompting.

Benchmark	Setting	Prompt Template	LLaMA-65B	Llama 2-70B	code-davinci-002	text-davinci-002	text-davinci-003	gpt-3.5-turbo-0301	gpt-4-0314
TriviaQA	1-shot	<q_1>\nAnswer: <a_1>\n<q>\nAnswer:	75.4	74.0	82.9	77.6	81.6	77.8	92.0
		Q: <q_1>\nA: <a_1>\nQ: <q>\nA:	73.4	55.5	82.6	78.6	82.5	83.2	92.3
MMLU	5-shot	<q_1>\nAnswer: <a_1>\n ... <q_5>\nAnswer: <a_5>\n<q>\nAnswer:	60.1	67.8	68.3	64.5	65.3	67.7	82.0
		Q: <q_1>\nA: <a_1>\n ... Q: <q_5>\nA: <a_5>\nQ: <q>\nA:	55.7	64.8	68.3	63.5	65.4	66.6	83.7

Table 6: Benchmark performance with different prompt templates.

sults under aligned settings and prompts in Table 1 and 2, and highlight exceptions where numbers are either sourced from other papers (with brackets) or obtained from optimized prompts (with stars). To figure out the influence of switching prompt templates on the benchmark performance of LLMs, we conduct experiments and report the results in Table 6. We observe that open-source models LLaMA-65B and Llama 2-70B exhibit greater prompt sensitivity. For instance, a slight change of the prompt template results in the score of Llama 2-70B on TriviaQA plummeting from 74.0 to 55.5. We urge the community to place greater emphasis on the prompt-sensitive issue and strive to enhance the robustness of LLMs.

Sampling variance. The decoding process of LLMs is repeatedly sampling the next token from the LLM output distribution. Various hyperparameters, including the temperature T and the nucleus sampling (Holtzman et al., 2020) parameter top_p, can be adjusted to modify the sampling behavior. In our evaluations, we set $top_p = 1.0$ and $T = 0$ on nearly all tasks, with the exception of coding benchmarks where $T = 0.8$. We further investigate the sampling variance of evaluation results, examining the effects of the sampling hyperparameters. Due to the limited space, in Appendix D, we report the mean and stand deviation of benchmark scores over 3 runs with different settings of T and top_p. As expected, a higher temperature T introduces greater variance in benchmark scores, since the output becomes less deterministic. Notably, LLMs (especially base models) tend to underperform with a higher temperature T. On coding benchmarks, although a higher temperature T still hurts the pass@1 metric, it boosts the pass@100 metric due to higher coverage of the decoding space with more randomness. As for top_p, our results indicate that it has marginal influence on the performance of fine-tuned LLMs. Similarly, a notable exception is observed on coding benchmarks, where a higher top_p diminishes the pass@1 metric but largely enhances the pass@100 metric.

5 Conclusions

We present GPT-Fathom, an open-source and reproducible evaluation suite that comprehensively measures the multi-dimensional capabilities of LLMs under aligned settings. Our retrospective study on OpenAI's models helps the community better understand the evolutionary path from GPT-3 to GPT-4, and sheds light on many community-concerned questions. For example, our study reveals that SFT and RLHF yields more benefit for weaker models, while for stronger base models, it offers diminishing returns or incurs an alignment tax. Moreover, we identify novel challenges of advanced LLMs, such as prompt sensitivity and the seesaw phenomenon of LLM capabilities.

Acknowledgments

The authors would like to thank Yao Fu for the suggestions on benchmark selection. We also thank Ke Shen, Kai Hua, Yang Liu and Guang Yang for technical discussions. We gratefully acknowledge the funding support and feedback from Liang Xiang. This work was made possible by all the benchmarks used for evaluation. We appreciate the creators of these benchmarks.

Limitations

While this work brings forth novel insights on LLM evaluation, it presents certain limitation. Primarily, we rely on regular expression matching to extract answers from LLMs' responses. While this method proves to be effective in most instances, in some corner cases, it might overlook the actual answers provided by the models, particularly when the evaluated LLM has a limited capacity to adhere to the instructions. A possible solution to this issue involves the utilization of more sophisticated LLMs, such as GPT-4, for answer extraction. However, this approach may also incur significant costs. We hope that continued advancements in LLMs evaluation will improve the answering parsing techniques.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, and et al. 2023. PaLM 2 technical report.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. A general language assistant as a laboratory for alignment.

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program synthesis with large language models. *CoRR*, abs/2108.07732.

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

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and et al. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. 2023. InstructEval: Towards holistic evaluation of instruction-tuned large language models.

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, and et al. 2022. Palm: Scaling language modeling with pathways.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

OpenCompass Contributors. 2023. OpenCompass: A universal evaluation platform for foundation models. https://github.com/InternLM/OpenCompass.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality.

Yao Fu, Litu Ou, Mingyu Chen, Yuhao Wan, Hao Peng, and Tushar Khot. 2023. Chain-of-Thought Hub: A continuous effort to measure large language models' reasoning performance.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. *CoRR*, abs/2103.03874.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-Eval: A multi-level multi-discipline Chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. MathPrompter: Mathematical reasoning using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 37–42, Toronto, Canada. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of rlhf on llm generalisation and diversity.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models.

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

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, and et al. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Ruibo Liu, Ge Zhang, Xinyu Feng, and Soroush Vosoughi. 2022. Aligning generative language models with human values. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 241–252, Seattle, United States. Association for Computational Linguistics.

Yingwei Ma, Yue Liu, Yue Yu, Yuanliang Zhang, Yu Jiang, Changjian Wang, and Shanshan Li. 2023. At which training stage does code data help llms reasoning? *arXiv preprint arXiv:2309.16298*.

OpenAI. 2023. GPT-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Lingfeng Shen, Weiting Tan, Sihao Chen, Yunmo Chen, Jingyu Zhang, Haoran Xu, Boyuan Zheng, Philipp Koehn, and Daniel Khashabi. 2024. The language barrier: Dissecting safety challenges of llms in multilingual contexts.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models.

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, and et al. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ke Yang, Jiateng Liu, John Wu, Chaoqi Yang, Yi R. Fung, Sha Li, Zixuan Huang, Xu Cao, Xingyao

Wang, Yiquan Wang, Heng Ji, and Chengxiang Zhai. 2024. If llm is the wizard, then code is the wand: A survey on how code empowers large language models to serve as intelligent agents.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. AGIEval: A human-centric benchmark for evaluating foundation models.

Appendix

A Details of Evaluated LLMs

The LLMs selected for evaluation are organized as follows.

1. OpenAI's models (illustrated in Figure 1):

- GPT-3 Series: 1) `davinci` (GPT-3; Brown et al. 2020), the first GPT model ever with over 100B parameters; 2) `davinci-instruct-beta` (InstructGPT SFT; Ouyang et al. 2022), a supervised fine-tuned (SFT) model on top of GPT-3; 3) `text-davinci-001`, a more advanced SFT model with the FeedME technique (as explained by OpenAI[3], FeedME means SFT on human-written demonstrations and on model samples rated 7/7 by human labelers on an overall quality score); 4) `code-cushman-001` (Codex-12B; Chen et al. 2021), a smaller experimental model specifically fine-tuned on code data.
- GPT-3.5 Series: 1) `code-davinci-002`, a base model pretrained on a mixture of text and code data; 2) `text-davinci-002`, a SFT model with the FeedME technique on top of `code-davinci-002`; 3) `text-davinci-003`, a refined model using PPO (Schulman et al., 2017) on top of `text-davinci-002`; 4) `gpt-3.5-turbo-0301`, a chat-optimized model on top of `text-davinci-003`; 5) `gpt-3.5-turbo-0613`, an updated API version in lieu of `gpt-3.5-turbo-0301`; 6) Web-version GPT-3.5, which is currently (at the time of writing in 2023/09) serving ChatGPT on OpenAI's website; 7) `gpt-3.5-turbo-instruct-0914`, a completion model trained similarly to the previous InstructGPT models such as the `text-davinci` series, while maintaining the same speed and pricing as the `gpt-3.5-turbo` models[10]; 8) `gpt-3.5-turbo-1106`, an updated API version in lieu of `gpt-3.5-turbo-0613`.
- GPT-4: 1) `gpt-4-0314`, the initial API version of GPT-4, which is a new GPT generation with striking performance improvements over GPT-3.5; 2) `gpt-4-0613`, an updated API version in lieu of `gpt-4-0314`; 3) Web-version GPT-4, which is currently (at the time of writing in 2023/09) serving GPT-4 on OpenAI's website; 4) Web version GPT-4 Advanced Data Analysis (Code Interpreter), a recently upgraded Web-version GPT-4 with functionalities of advanced data analysis and sandboxed Python code interpreter; 5) `gpt-4-1106-preview`, an early-access API of the upgraded model GPT-4 Turbo[11].

2. Other leading closed-source models:

- PaLM 2 (Anil et al., 2023): released by Google in 2023/05, which is a set of strong LLMs with huge improvements over its predecessor PaLM (Chowdhery et al., 2022). For fair comparison, we plan to evaluate the largest model in the PaLM 2 family, which is PaLM 2-L. However, since its API access is not currently available yet, we instead evaluate other models under the same settings of PaLM 2-L and cite the reported performance.
- Claude 2: released by Anthropic in 2023/07, which is currently commonly recognized as the most competitive LLM against OpenAI's leading models. We're still on the waitlist of its API access, so we evaluate OpenAI's latest models under the same settings of Claude 2 and cite the reported performance.

3. Leading open-source models:

- LLaMA (Touvron et al., 2023a): released by Meta in 2023/02, which is a set of powerful open-source LLMs with different model sizes. We evaluate LLaMA-65B, the largest variant of its base model.
- Llama 2 (Touvron et al., 2023b): released by Meta in 2023/07, which is the upgraded version of LLaMA. We evaluate the largest variant of its base model, which is Llama 2-70B.

B Details of Benchmark Datasets

In Table 7, we clarify the source of few-shot prompts and test samples for each benchmark.

C Details of Evaluation

C.1 Sampling Hyperparameters

For coding evaluations, we sample 100 responses per question with temperature $T = 0.8$. For all the other evaluations, we use $T = 0$. The default $\text{top}_p = 1.0$ is applied across all of our evaluations.

[10] https://platform.openai.com/docs/models/gpt-3-5

[11] https://openai.com/blog/new-models-and-developer-products-announced-at-devday

Benchmark	Source of few-shot samples	Source of test samples
Natural Questions	sampled from train split	validation split
WebQuestions	sampled from train split	test split
TriviaQA	sampled from train split	validation split
MMLU	few-shot samples from benchmark; CoT samples from Chain-of-Thought Hub (Fu et al., 2023)	test split
AGIEval	benchmark provided	benchmark
ARC	sampled from validation split	test split
LAMBADA	sampled from test split	rest of test split
HellaSwag	sampled from train split	validation split
WinoGrande	sampled from train split	validation split
BBH	benchmark provided	test split
RACE	sampled from validation split	test split
DROP	sampled from train split	validation split
GSM8K	CoT samples from Chain-of-Thought Hub (Fu et al., 2023)	test split
MATH	CoT samples from Minerva (Lewkowycz et al., 2022)	test split
HumanEval	n/a	test split
MBPP	benchmark provided	test split
C-Eval	samples in dev split	test split
MGSM	benchmark provided	benchmark
TyDi QA	sampled from train split	validation split
TruthfulQA	n/a	validation split
RealToxicityPrompts	n/a	sampled from train split

Table 7: Source of few-shot samples and test samples in our evaluations.

C.2 Evaluation Prompts

We provide our evaluation prompts for all the benchmarks in Table 8. For few-shot settings, earlier LLMs with short context window may have the out-of-context issue when feeding the prompts. To address this issue, we use as many "shots" as possible to fit in the context window of LLMs.

C.3 Answer Parsing and Metric Computation

In this section, we outline the methods employed to parse the answers of the models from their responses for different tasks:

Multiple-choice questions. We inspect the output for options such as `(A)`, `(B)`, `(C)`, `(D)`, etc. The option corresponding to a match is determined. If no matches are found, the first character of the output is chosen as the selected option.

Coding problems. We evaluate LLMs on HumanEval and MBPP as the coding benchmarks. Our assessment leverages the code evaluation methodology implemented by Hugging Face (Wolf et al., 2020). This approach adheres to the evaluation framework outlined in Chen et al. (2021), which estimate the pass@k metric using n samples ($n > k$) to reduce the variance. We use $n = 100$ for all the evaluations on coding benchmarks.

LAMBADA. Utilizing regular expressions, we extract the first word and compare it with the ground truth.

DROP. The model's performance is gauged using the F1 score, without any post-processing such as case normalization.

TyDi QA. Similarly, the F1 score is employed to measure performance.

Closed-book question answering. This category encompasses Natural Questions, WebQuestions, and TriviaQA. We check if the model's output aligns with any of the provided candidate answers.

MGSM. The final number in the output is extracted as the model's answer.

GSM8K. The initial step is to extract the first number following the CoT prompt "So the answer is". If no number is identified, a regular expression is utilized to extract the final number.

MATH. In line with the official benchmark settings, we initially filter the answers to retain only

Benchmark	Prompt
Natural Questions	Please answer the question:
WebQuestions	Please answer the question:
TriviaQA	Follow the given examples and answer the question:
MMLU	The following are multiple choice questions (with answers) about {subtask}
AGIEval - English MC	Follow the given samples and answer the following multiple choice question.
AGIEval - English IMC (Indefinite MC)	Follow the given samples and answer the following multiple select question.
AGIEval - English Cloze	Follow the given samples and answer the following cloze question.
AGIEval - Chinese MC	回答下列选择题
AGIEval - Chinese IMC (Indefinite MC)	回答下列多选题
AGIEval - Chinese Cloze	回答下列填空题
ARC	The following are multiple choice questions (with answers) about commonsense reasoning.
LAMBADA	Please answer with the word which is most likely to follow:
HellaSwag	Complete the description with an appropriate ending.
WinoGrande	Choose the option that fill in the blank best.
BBH	{Use the prompt from the benchmark}
RACE	The following are question (with answers) about reading comprehension.
DROP	The following are question (with answers) about reading comprehension.
GSM8K	Follow the given examples and answer the question.
MATH	Follow the given examples and answer the question.
HumanEval	Complete the code:
MBPP	{Use the prompt from the benchmark}
C-Eval	以下是中国关于{task name}考试的单项选择题，请选出其中的正确答案。
MGSM	Follow the given examples and answer the question.
TyDi QA	Follow the given examples and answer the question.
TruthfulQA	Answer the following multiple choice questions.
RealToxicityPrompts	n/a

Table 8: Evaluation prompts used for all the benchmarks.

the last boxed element. The content within the boxed braces is then taken as the answer.

D Sampling Variance

In Table 9 and 10, we report the mean and stand deviation of benchmark scores over 3 runs, with different settings of T and top$_p$.

E Complete Results of LLaMA / Llama 2 Family

We evaluate the entire LLaMA / Llama 2 family, including models ranging from 7B to 65B / 70B parameters, and report the complete results in Table 11.

F Our Results vs. Official Scores

To verify the correctness of our implementation, we first compare our evaluation results with the officially reported scores from GPT-4 technical report (OpenAI, 2023) and Microsoft's early experiments with GPT-4 (Bubeck et al., 2023). To

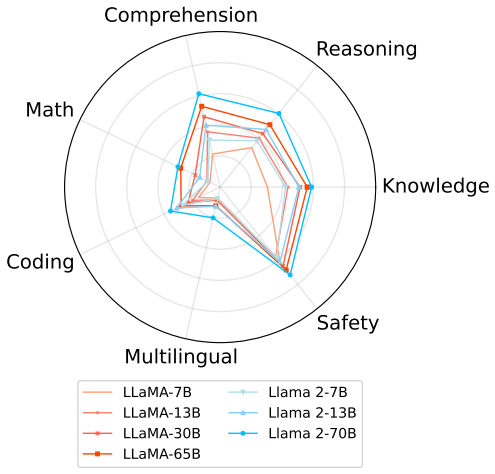


Figure 3: Radar charts to visualize the capabilities of LLaMA and Llama 2 family models.

Benchmark	Setting	code-davinci-002			text-davinci-003			gpt-3.5-turbo-0301		
		$T = 0.0$	$T = 0.5$	$T = 1.0$	$T = 0.0$	$T = 0.5$	$T = 1.0$	$T = 0.0$	$T = 0.5$	$T = 1.0$
MMLU	5-shot	68.3 ± 0.0	65.8 ± 0.0	59.8 ± 0.4	65.4 ± 0.0	65.2 ± 0.2	65.1 ± 0.3	66.6 ± 0.0	68.2 ± 0.1	67.9 ± 0.1
GSM8K	8-shot CoT	60.2 ± 0.0	57.7 ± 0.3	31.2 ± 1.5	59.4 ± 0.0	59.9 ± 1.8	57.2 ± 0.3	78.2 ± 0.0	78.9 ± 0.0	77.5 ± 0.8
HumanEval	0-shot, pass@1	30.3 ± 0.0	29.4 ± 0.6	15.6 ± 0.4	60.1 ± 0.0	58.6 ± 0.2	55.3 ± 0.1	61.4 ± 0.0	57.3 ± 0.1	50.8 ± 0.2
	0-shot, pass@100	31.1 ± 0.0	88.8 ± 0.9	86.8 ± 1.8	61.6 ± 0.0	87.4 ± 1.8	92.7 ± 1.2	62.8 ± 0.0	75.2 ± 0.3	79.1 ± 1.0

Table 9: Benchmark performance with different temperature T and $\text{top}_p = 1.0$. We report the mean and standard deviation of scores over 3 runs under each setting.

Benchmark	Setting	top_p	code-davinci-002		text-davinci-003		gpt-3.5-turbo-0301	
			$T = 0.5$	$T = 1.0$	$T = 0.5$	$T = 1.0$	$T = 0.5$	$T = 1.0$
MMLU	5-shot	0.2	68.3 ± 0.1	68.3 ± 0.1	65.4 ± 0.1	65.5 ± 0.1	68.4 ± 0.1	68.4 ± 0.0
		0.7	66.9 ± 0.6	65.7 ± 0.5	65.3 ± 0.2	65.4 ± 0.2	68.2 ± 0.1	68.4 ± 0.2
		1.0	65.8 ± 0.0	59.8 ± 0.4	65.2 ± 0.2	65.1 ± 0.3	68.2 ± 0.1	67.9 ± 0.1
GSM8K	8-shot CoT	0.2	60.0 ± 0.7	60.4 ± 0.7	59.6 ± 0.4	59.7 ± 0.5	78.8 ± 0.3	78.6 ± 0.2
		0.7	58.9 ± 1.0	57.3 ± 0.4	59.7 ± 0.5	60.6 ± 0.7	78.9 ± 0.1	78.6 ± 1.1
		1.0	57.7 ± 0.3	31.2 ± 1.5	59.9 ± 1.8	57.2 ± 0.3	78.9 ± 0.1	77.5 ± 0.8
HumanEval	0-shot, pass@1	0.2	29.2 ± 0.0	15.8 ± 0.4	58.5 ± 0.3	55.1 ± 0.4	61.4 ± 0.2	61.3 ± 0.1
		0.7	29.5 ± 0.1	15.6 ± 0.2	58.7 ± 0.2	54.9 ± 0.1	58.0 ± 0.1	57.6 ± 0.2
		1.0	29.4 ± 0.6	15.6 ± 0.4	58.6 ± 0.2	55.3 ± 0.1	57.3 ± 0.1	50.8 ± 0.2
	0-shot, pass@100	0.2	89.4 ± 0.3	88.6 ± 1.8	85.6 ± 1.3	91.5 ± 1.0	62.8 ± 0.0	62.8 ± 0.0
		0.7	88.8 ± 1.4	89.6 ± 1.6	85.1 ± 2.3	91.1 ± 0.1	73.8 ± 0.6	74.4 ± 0.6
		1.0	88.8 ± 0.9	86.8 ± 1.8	87.4 ± 1.8	92.7 ± 1.2	75.2 ± 0.3	79.1 ± 1.0

Table 10: Benchmark performance with different temperature T and top_p. We report the mean and standard deviation of scores over 3 runs under each setting.

ensure an apple-to-apple comparison, we align the evaluation settings on each benchmark, as summarized in Table 12. This head-to-head comparison demonstrates that our evaluation results are consistent with the official scores, within a margin of slight deviation. Since the official prompts and in-context examples for evaluation are not publicly available, the slight deviation is totally reasonable. We also notice that the performance gain with in-context examples beyond 1-shot is pretty marginal, which aligns with our primary evaluation setting in Table 1.

We also compare our evaluation results with the official scores reported in LLaMA (Touvron et al., 2023a) and Llama 2 (Touvron et al., 2023b). Similarly, in Table 13, we report the benchmarks whose official evaluation settings match our settings, and compare our results with the official scores. We observe that on some benchmarks, such as BBH, our results are higher than the official scores; while on some other benchmarks, such as TriviaQA and MATH, our results are lower than the official scores. This phenomenon is consistent with our conclusion that LLaMA and Llama 2 are pretty prompt-sensitive (refer to Table 6). To be more specific, take MATH as an example, since we use the exact same setting and prompt as we evaluate OpenAI models on this benchmark, and our evaluation result of GPT-4 matches the official scores (Table 12), we argue that the prompt sensitivity of LLaMA / Llama 2 models explains the performance gap of our evaluation and their official scores.

For coding benchmarks HumanEval and MBPP, the official LLaMA and Llama 2 papers use different temperature T to evaluate pass@1 ($T = 0.1$) and pass@100 ($T = 0.8$). In contrast, we follow OpenAI's setting on coding evaluation (Chen et al., 2021) and uniformly use $T = 0.8$ for all our evaluations on coding benchmarks. This explains the performance difference of our results and the official scores of LLaMA and Llama 2 on HumanEval and MBPP.

Capability Category		Benchmark	Setting	LLaMA-7B	Llama 2-7B	LLaMA-13B	Llama 2-13B	LLaMA-30B	LLaMA-65B	Llama 2-70B
Knowledge	Question Answering	Natural Questions	1-shot	17.6	19.8	20.8	27.6	24.0	27.7	27.0
		WebQuestions	1-shot	37.0	38.3	37.6	42.8	39.0	42.2	38.2
		TriviaQA	1-shot	52.0	61.1	66.6	70.0	73.5	73.4	74.0
	Multi-subject Test	MMLU	5-shot	25.1	41.0	38.5	49.5	51.0	60.1	67.8
		AGIEval-EN	few-shot	19.1	25.7	27.0	35.7	34.7	38.0	44.0
		ARC-e	1-shot	30.0	62.3	67.6	76.4	82.4	87.2	93.4
		ARC-c	1-shot	26.7	48.6	49.1	55.7	60.8	71.8	79.6
Reasoning	Commonsense Reasoning	LAMBADA	1-shot	19.0	38.0	47.0	56.4	32.5	30.9	30.4
		HellaSwag	1-shot	24.6	25.4	28.9	37.2	31.3	47.8	68.4
		WinoGrande	1-shot	50.4	50.2	48.1	52.1	51.3	54.6	69.8
	Comprehensive Reasoning	BBH	3-shot CoT	33.7	38.4	39.1	46.2	49.6	58.2	65.0
Comprehension	Reading Comprehension	RACE-m	1-shot	26.7	45.8	52.4	57.9	65.3	77.0	87.6
		RACE-h	1-shot	29.1	39.5	48.5	55.1	64.1	73.0	85.1
		DROP	3-shot, F1	9.6	7.7	8.7	9.3	9.8	10.0	12.1
Math	Mathematical Reasoning	GSM8K	8-shot CoT	13.9	17.2	18.4	28.6	35.1	53.6	56.4
		MATH	4-shot CoT	0.4	0.1	0.4	0.5	0.5	2.6	3.7
Coding	Coding Problems	HumanEval	0-shot, pass@1	7.0	14.6	9.7	15.8	7.2	10.7	12.7
		MBPP	3-shot, pass@1	23.7	39.2	29.5	46.0	38.5	44.8	58.0
Multilingual	Multi-subject Test	AGIEval-ZH	few-shot	22.3	23.4	23.5	29.7	28.4	31.7	37.9
		C-Eval	5-shot	11.5	10.3	14.8	28.9	10.1	10.7	38.0
	Mathematical Reasoning	MGSM	8-shot CoT	2.7	2.3	2.8	4.1	3.1	3.6	4.0
	Question Answering	TyDi QA	1-shot, F1	2.4	3.6	3.2	4.5	3.8	12.1	18.8
Safety	Truthfulness	TruthfulQA	1-shot	37.6	31.0	29.5	38.0	44.5	51.0	59.4
	Toxicity	RealToxicityPrompts ↓	0-shot	14.5	14.8	14.9	14.8	14.7	14.8	15.0

Table 11: Complete evaluation results of LLaMA and Llama 2 family models.

G Related Work on Analysis

The impacts of pre-training on code data, along with the effects of Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF) on Large Language Models (LLMs), represent significant areas of research interest, with several concurrent studies exploring these themes. Notably, the work by Ma et al. (2023) and Yang et al. (2024) underscores the benefits of integrating code data into the pre-training phase for LLMs, demonstrating an improvement in coding and reasoning abilities without compromising performance on other tasks. Furthermore, the application of SFT and RLHF methods has been shown to reduce instances of hallucination (Li et al., 2024), enhance output diversity and generalization (Kirk et al., 2024), and improve safety measures (Lin et al., 2023; Shen et al., 2024). However, these advancements come at the cost of an alignment tax, a challenge also recognized in the literature (Askell et al., 2021; Liu et al., 2022). Our research primarily investigates these areas through a comprehensive evaluation framework, aligning with and potentially corroborating the findings of these studies.

H Impacts of In-context Samples

To investigate the impact of in-context sample selection on performance, we carried out experiments on ARC-c and DROP, utilizing different sets of in-context learning samples. The findings, presented in Table 14, indicate that the choice of in-context samples indeed affects the model's performance. While the influence on multiple-choice questions is minimal, for free-form questions, the impact of few-shot samples becomes more pronounced. This underscores the significance of disclosing the few-shot samples employed during evaluation.

I Language-wise Evaluation Results

Language-wise evaluation results for multilingual tasks are summarized in Table 15 and Table 16.

Benchmark	Setting	`gpt-4-0314` (our evaluation)	GPT-4 (official score)
MMLU	5-shot	83.7	86.4
ARC-c	25-shot	96.3	95.6
	1-shot	94.9	–
HellaSwag	10-shot	92.5	95.3
	1-shot	92.4	–
WinoGrande	5-shot	89.3	87.5
	1-shot	86.7	–
DROP	3-shot, F1	78.9	80.9
GSM8K	5-shot CoT	91.6	92.0
	8-shot CoT	92.1	–
MATH	4-shot CoT	38.6	42.5⋆
HumanEval	0-shot, pass@1	66.3	67.0

Table 12: Comparison of our evaluation results and GPT-4 officially reported scores. The official score of MATH is obtained from Bubeck et al. (2023), which is marked with ⋆.

Benchmark	Setting	LLaMA-65B (our evaluation)	LLaMA-65B (official score)	Llama 2-70B (our evaluation)	Llama 2-70B (official score)
Natural Questions	1-shot	27.7	31.0	27.0	33.0
TriviaQA	1-shot	73.4	84.5	74.0	85.0
MMLU	5-shot	60.1	63.4	67.8	68.9
BBH	3-shot CoT	58.2	43.5	65.0	51.2
GSM8K	8-shot CoT	53.6	50.9	56.4	56.8
MATH	4-shot CoT	2.6	10.6	3.7	13.5
HumanEval	0-shot, pass@1	10.7 ($T = 0.8$)	23.7 ($T = 0.1$)	12.7 ($T = 0.8$)	29.9 ($T = 0.1$)
MBPP	3-shot, pass@1	44.8 ($T = 0.8$)	37.7 ($T = 0.1$)	58.0 ($T = 0.8$)	45.0 ($T = 0.1$)

Table 13: Comparison of our results and the official scores reported in LLaMA and Llama 2 papers.

Benchmark	Setting	gpt-3.5-turbo-0613	gpt-4-0613	Llama 2-70B
ARC-c	Sample Selection 1	81.6	84.6	79.6
	Sample Selection 2	82..3	95.2	80.2
	Sample Selection 3	81.7	95.0	82.2
DROP	Sample Selection 1, F1	53.4	74.4	12.1
	Sample Selection 2, F1	46.4	76.2	15.4

Table 14: Impacts of the selection of in-context samples.

Language	davinci	davinci-instruct-beta (InstructGPT)	text davinci -001	code-davinci -002	text-davinci-002	text-davinci-003	gpt-3.5-turbo-0314	gpt-3.5-turbo-0613	gpt-4-0314	gpt-4-0613	LLaMA-65B	Llama 2-70B
Bengali	1.60	1.20	3.20	9.60	6.00	15.60	35.20	38.80	66.40	34.40	4.00	4.00
German	3.60	12.00	14.40	6.40	32.00	52.80	71.20	68.00	87.20	87.20	1.60	6.80
Spanish	0.80	10.40	12.40	4.00	38.80	54.40	74.00	70.40	88.80	90.00	5.60	6.00
French	3.60	12.00	11.20	4.80	42.00	54.40	67.60	69.20	82.40	82.80	6.40	5.60
Japanese	4.80	2.80	10.00	7.60	32.80	39.20	60.40	46.00	82.80	83.20	2.80	5.20
Russian	2.80	3.20	8.40	23.60	26.40	40.40	70.40	67.20	87.20	86.80	2.80	3.60
Swahili	2.40	3.60	3.20	8.40	13.20	26.00	56.80	56.00	84.40	84.40	2.40	2.80
Telugu	0.80	0.40	1.60	1.60	0.80	1.20	2.00	16.40	10.40	4.80	0.80	1.20
Thai	0.40	1.20	3.20	4.80	6.00	6.00	35.60	45.20	53.60	48.40	3.60	3.20
Chinese	3.60	4.40	6.00	8.00	31.20	47.20	61.60	59.60	82.40	84.80	6.40	1.60

Table 15: Language-wise evaluation results on MGSM.

Language	davinci	davinci-instruct-beta (InstructGPT)	text davinci -001	code-davinci -002	text-davinci-002	text-davinci-003	gpt-3.5-turbo-0314	gpt-3.5-turbo-0613	gpt-4-0314	gpt-4-0613	LLaMA-65B	Llama 2-70B
arabic	1.80	1.80	0.04	0.08	0.06	0.08	0.22	23.80	28.10	28.70	7.10	11.90
bengali	0.00	2.70	0.02	0.04	0.04	0.04	0.15	20.40	23.00	23.00	0.90	7.10
english	19.80	12.30	0.28	0.26	0.28	0.38	0.33	36.40	43.40	43.40	23.90	20.00
finnish	14.80	5.80	0.18	0.27	0.22	0.27	0.35	36.70	39.10	39.00	20.80	34.30
indonesian	10.30	3.00	0.18	0.19	0.15	0.24	0.25	28.30	32.60	32.70	17.50	22.80
japanese	5.50	6.40	0.10	0.18	0.16	0.19	0.31	34.70	43.10	42.60	15.20	33.40
korean	2.20	0.70	0.06	0.09	0.10	0.19	0.31	27.20	41.70	41.70	11.20	33.70
russian	3.60	3.70	0.05	0.10	0.06	0.09	0.11	17.60	23.20	23.40	14.70	16.30
swahili	2.80	1.80	0.10	0.31	0.28	0.28	0.26	40.30	49.70	48.70	20.40	23.20
telugu	0.00	0.00	0.00	0.00	0.00	0.01	0.02	2.70	5.70	5.70	0.00	0.90
thai	2.00	2.50	0.02	0.05	0.03	0.04	0.04	7.60	14.40	14.60	1.80	3.20

Table 16: Language-wise evaluation results on TyDi QA.