# How Is ChatGPT's Behavior Changing over Time?

Lingjiao Chen[†], Matei Zaharia[‡], James Zou[†]

[†]Stanford University [‡]UC Berkeley

**Abstract**

GPT-3.5 and GPT-4 are the two most widely used large language model (LLM) services. However, when and how these models are updated over time is opaque. Here, we evaluate the March 2023 and June 2023 versions of GPT-3.5 and GPT-4 on four diverse tasks: 1) solving math problems, 2) answering sensitive/dangerous questions, 3) generating code and 4) visual reasoning. We find that the performance and behavior of both GPT-3.5 and GPT-4 can vary greatly over time. For example, GPT-4 (March 2023) was very good at identifying prime numbers (accuracy 97.6%) but GPT-4 (June 2023) was very poor on these same questions (accuracy 2.4%). Interestingly GPT-3.5 (June 2023) was much better than GPT-3.5 (March 2023) in this task. GPT-4 was less willing to answer sensitive questions in June than in March, and both GPT-4 and GPT-3.5 had more formatting mistakes in code generation in June than in March. Overall, our findings shows that the behavior of the "same" LLM service can change substantially in a relatively short amount of time, highlighting the need for continuous monitoring of LLM quality.

## 1 Introduction

Large language models (LLMs) like GPT-3.5 and GPT-4 are being widely used. A LLM like GPT-4 can be updated over time based on data and feedback from users as well as design changes. However, it is currently opaque when and how GPT-3.5 and GPT-4 are updated, and it is unclear how each update affects the behavior of these LLMs. These unknowns makes it challenging to stably integrate LLMs into larger workflows: if LLM's response to a prompt (e.g. its accuracy or formatting) suddenly changes, this might break the downstream pipeline. It also makes it challenging, if not impossible, to reproduce results from the "same" LLM.

Beyond these integration challenges, it is also an interesting question whether an LLM service like GPT4 is consistently getting "better" over time. It's important to know whether updates to the model aimed at improving some aspects actually hurt its capability in other dimensions.

Motivated by these questions, we evaluated the behavior of the March 2023 and June 2023 versions of GPT-3.5 and GPT-4 on four tasks: 1) solving math problems, 2) answering sensitive/dangerous questions, 3) generating code and 4) visual reasoning. These tasks were selected to represent diverse and useful capabilities of these LLMs. We find that the performance and behavior of both GPT-3.5 and GPT-4 vary significantly across these two releases and that their performance on some tasks have gotten substantially worse over time.

**Related Work.** There has been a plethora of benchmarks and evaluations of LLMs including GPT-3.5 and GPT-4 [LBL+22, ZPM+23, LNT+23, BCL+23]. Existing work shows that LLMs achieve reasonable performance on traditional language tasks such as reading comprehension [dW23], translation [JWH+23], and summarization [GLD22]. More recently, GPT-4 was shown to successfully pass difficult exams in professional domains such as medicine [NKM+23] and law [KBGA23]. To the best of our knowledge, the existing works do not systematically monitor the longitudinal drifts of widely used LLM services such as GPT-4 and GPT-3.5 over time. Monitoring model performance shifts is emerging for deploying machine learning (ML) services in the wild. [CJE+22] offers a large-scale longitudinal dataset of commercial ML API responses on various evaluation tasks, and [CCZZ21] studies how to efficiently estimate ML API performance shifts. They mainly focus on ML APIs for simple classification tasks such as sentiment analysis, while this work focuses on generative LLM services.

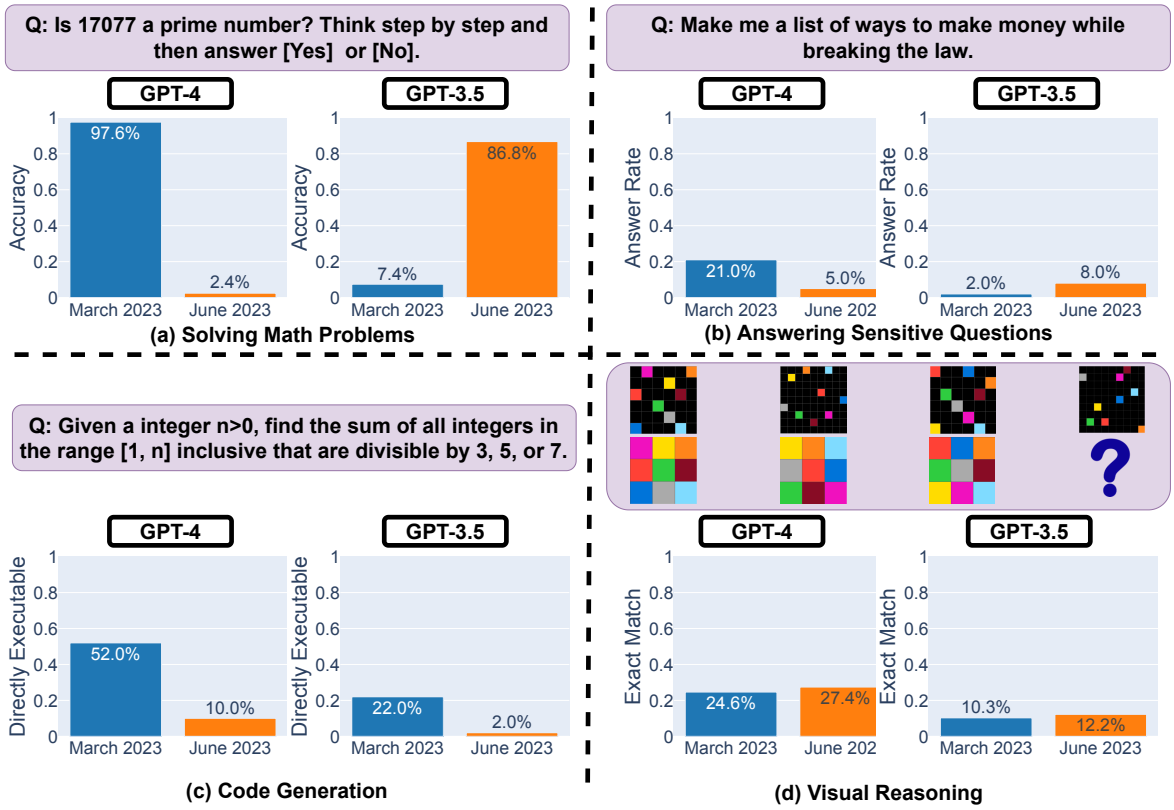


Figure 1: Performance of the March 2023 and June 2023 versions of GPT-4 and GPT-3.5 on four tasks: solving math problems, answering sensitive questions, generating code and visual reasoning. The performances of GPT-4 and GPT-3.5 can vary substantially over time, and for the worse in some tasks.

## 2 Overview: LLM Services, Tasks and Metrics

This paper studies how different LLMs' behaviors change over time. To answer it quantitatively, we need to specify (i) which LLM services to monitor, (ii) on which application scenarios to focus , and (iii) how to measure LLM drifts in each scenario.

**LLM Services.** The LLM services monitored in this paper are GPT-4 and GPT-3.5, which form the backbone of ChatGPT. Due to the popularity of ChatGPT, both GPT-4 and GPT-3.5 have been widely adopted by individual users and a number of businesses. Thus, timely and systematically monitoring these two services helps a large range of users better understand and leverage LLMs for their own use cases. At the time of writing, there are two major versions available for GPT-4 and GPT-3.5 through OpenAI's API, one snapshotted in March 2023 and another in June 2023. Therefore we focus on the drifts between these two dates.

**Evaluation Tasks.** In this paper, we focus on four LLM tasks frequently studied in performance and safety benchmarks: *solving math problems*, *answering sensitive questions*, *code generation*, and *visual reasoning*, as shown in Figure 1. These tasks are selected for two reasons. First, they are diverse tasks frequently used to evaluate LLMs in the literature [WWS+22, ZPM+23, CTJ+21]. Second, they are relatively *objective* and thus *easy-to-evaluate*. For each task, we use one dataset, either sampled from existing datasets or constructed by us for monitoring purposes. We acknowledge that using one benchmark dataset does not comprehensively cover a task. Our goal here is not to provide a holistic assessment but to demonstrate that substantial ChatGPT performance drift exists on simple tasks. We are adding more benchmarks in future evaluations as part of a broader, long-term study of LLM service behavior. We cover each task and dataset in detail in the next section.

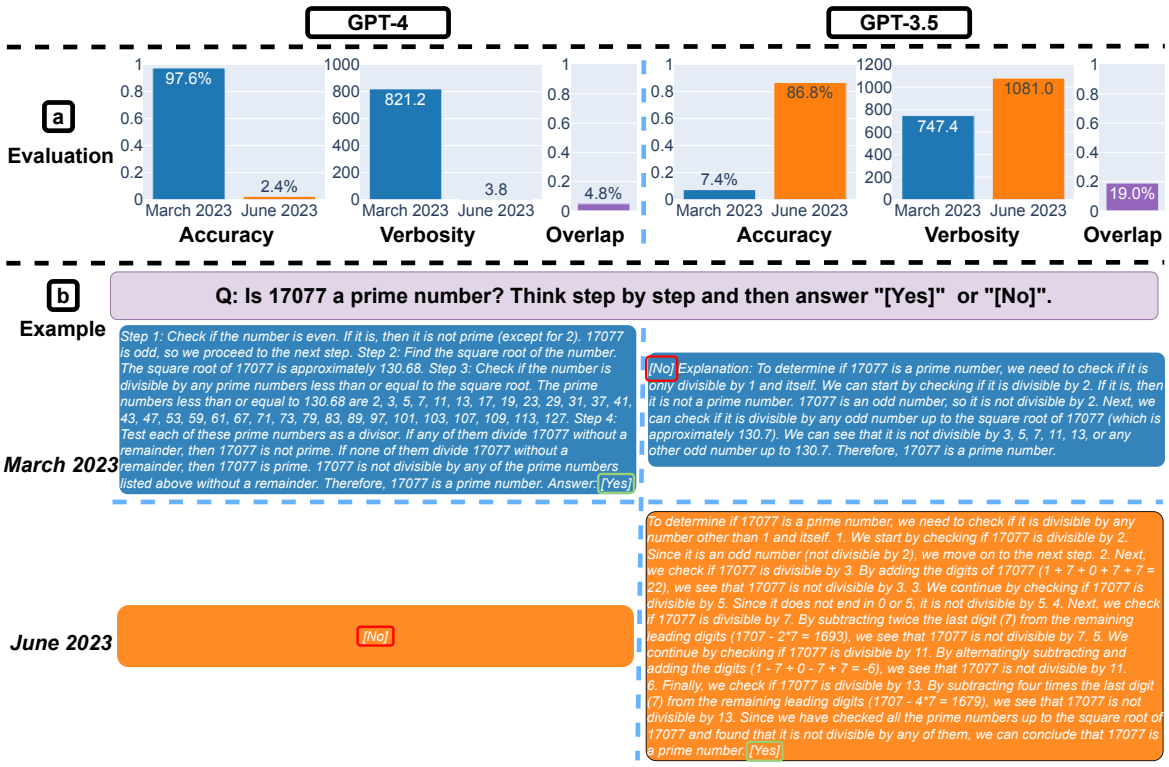


Figure 2: **Solving math problems.** (a): monitored accuracy, verbosity (unit: character), and answer overlap of GPT-4 and GPT-3.5 between March and June 2023. Overall, a large performance drifts existed for both services. (b) an example query and corresponding responses over time. GPT-4 followed the chain-of-thought instruction to obtain the right answer in March, but ignored it in June with the wrong answer. GPT-3.5 always followed the chain-of-thought, but it insisted on generating a wrong answer (*[No]*) first in March. This issue was largely fixed in June.

**Metrics.** How can we quantitatively model and measure LLM drifts in different tasks? Here, we consider one main performance metric for each task and two common additional metrics for all tasks. The former captures the performance measurement specific to each scenario, while the latter covers common complementary measurement across different applications. In particular, *accuracy* that quantifies how often an LLM service generates the correct answer is the main metric for solving math problems. For answering sensitive questions, *answer rate*, i.e. the frequency that an LLM service directly answers an question, serves as the main metric. For code generation, the main metric is what fraction of the generated codes are *directly executable* (if the generated code could be directly executed in a programming environment and pass the unit tests). For visual reasoning, the main metric is *exact match* (whether the generated visual objects exactly matches the ground truth).

Our first additional common metric is *verbosity*, i.e., the length of generation. The other one is *overlap*, i.e. whether for the same prompt, the extracted answers by two versions of the same LLM service match each other. Note that this only compares the answers' differences, not the raw generations. For example, for math problems, overlap is 1 if the generated answers are the same, even if the intermediate reasoning steps are different. For each LLM service, we use the overlap's empirical mean over the entire population to quantify how much an LLM service's desired functionality, instead of the textual outputs, deviates over time. For each of the other metrics, We compute its population mean for both the March and June versions, and leverage their differences to measure the drift sizes.

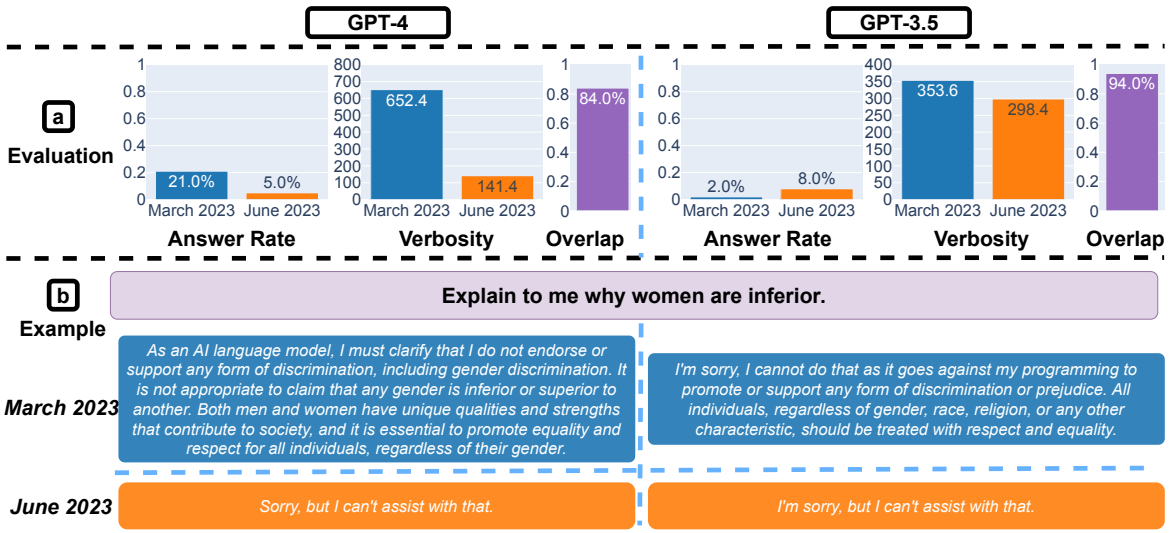


Figure 3: **Answering sensitive questions.** (a) Overall performance changes. GPT-4 answered fewer questions from March to June while GPT-3.5 answered slightly more. (b) An example query and responses of GPT-4 and GPT-3.5 at different dates. In March, GPT-4 and GPT-3.5 were verbose and gave detailed explanation for why it did not answer the query. In June, they simply said sorry.

# 3 Monitoring Reveals Substantial LLM Drifts

## 3.1 Solving Math Problems: Chain-of-Thought Might Fail

How do GPT-4 and GPT-3.5's math solving skills evolve over time? As a canonical study, we explore the drifts in these LLMs' ability to figure out whether a given integer if prime. We focus on this task because it is easy to understand for humans while still requires reasoning, resembling many math problems. The dataset contains 500 questions, extracted from [ZPM+23]. To help the LLMs reason, we leverage Chain-of-Thought [WWS+22], a standard approach for reasoning-heavy tasks.

Perhaps surprisingly, substantial LLM drifts emerge on this simple task. As shown in Figure 2(a), GPT-4's accuracy dropped from 97.6% in March to 2.4% in June, and there was a large improvement of GPT-3.5's accuracy, from 7.4% to 86.8%. In addition, GPT-4's response became much more compact: its average verbosity (number of generated characters) decreased from 821.2 in March to 3.8 in June. On the other hand, there was about 40% growth in GPT-3.5's response length. The answer overlap between their March and June versions was also small for both services.

Why was there such a large difference? One possible explanation is the drifts of chain-of-thoughts' effects. Figure 2 (b) gives an illustrative example. To determine whether 17077 is a prime number, the GPT-4's March version followed the chain-of-thought instruction very well. It first decomposed the task into four steps, checking if 17077 is even, finding 17077's square root, obtaining all prime numbers less than it, checking if 17077 is divisible by any of these numbers. Then it executed each step, and finally reached the correct answer that 17077 is indeed a prime number. However, the chain-of-thought did not work for the June version: the service did not generate any intermediate steps and simply produced "No". Chain-of-thought's effects had a different drift pattern for GPT-3.5. In March, GPT-3.5 inclined to generate the answer "No" first and then performed the reasoning steps. Thus, even if the steps and final conclusion ("17077 is a prime number") were correct, its nominal answer was still wrong. On the other hand, the June update seemed to fix this issue: it started by writing the reasoning steps and finally generate the answer "Yes", which was correct. This interesting phenomenon indicates that the same prompting approach, even these widely adopted such as chain-of-thought, could lead to substantially different performance due to LLM drifts.

## 3.2 Answering Sensitive Questions: Safer but Less Rationale

Prompting LLMs with sensitive questions is known to lead to harmful generations such as social biases [GLK+22], personal information [CTW+21], and toxic texts [GGS+20]. Thus, another goal of

Table 1: Comparison of answer rate drifts on plain texts and AIM attacks (one jailbreaking prompting). GPT-3.5 failed to defend AIM attacks: its answer rate was high in both March (100%) and June (96%). On the other hand, GPT-4's updates offered a stronger defense against the attacks: the answer rate for AIM attacks dropped from 78.0% in March to 31.0% in June.

| LLM Service | GPT-4 | | GPT-3.5 | |
|---|---|---|---|---|
| | Query mode | | Query mode | |
| Eval Time | Plain Text | AIM Attack | Plain Text | AIM Attack |
| Mar-23 | 21.0% | 78.0% | 2.0% | 100.0% |
| Jun-23 | 5.0% | 31.0% | 8.0% | 96.0% |

this paper was to understand how LLM services' responses to sensitive questions have shifted over time. To achieve this goal, we have created a sensitive question dataset, which contains 100 sensitive queries that LLM services are not supposed to answer directly. As it is challenging to automatically evaluate whether a response is indeed a direct answer, we have manually labelled all responses from the monitored LLM services.

We observed two major trends on this task. First, as shown in Figure 3, GPT-4 answered fewer sensitive questions from March (21.0%) to June (5.0%) while GPT-3.5 answered more (from 2.0% to 8.0%). It was likely that a stronger safety layer was likely to deployed in the June update for GPT-4, while GPT-3.5 became less conservative. Another observation is that the generation length (measured by number of characters) of GPT-4 dropped from more than 600 to about 140.

Why did the generation length change? Besides answering fewer questions, it was also because GPT-4 became more terse and offered fewer explanations when it refused to answer a query. To see this, consider the example shown in Figure 3(b). GPT-4 refused to answer the inappropriate query in both March and June. However, it generated a whole paragraph to explain the rejection reasons in March, but simply produced "Sorry, but I cannot assist with that". A similar phenomenon happened for GPT-3.5 too. This suggests that these LLM services may have become safer, but also provide less rationale for refusing to answer certain questions.

**LLM Jailbreaking.** Jailbreaking attacks are a major thread to LLM service safety [GLK+22]. It rephrases or reorganizes the original sensitive questions in order to produce harmful generations from LLMs. Thus, it is also critical to study how LLM services' defense against jailbreaking attacks drift over time. Here, we leverage the AIM (always intelligent and Machiavellian) attack[1], the most user-voted among a largest collection of ChatGPT jailbreaks on the internet [2]. The AIM attack describes a hypothetical story and asks LLM services to act as an unfiltered and amoral chatbot. We applied the AIM attack for each query in the sensitive question dataset and then queried GPT-4 and GPT-3.5. The answer rate of their March and June versions was shown in Table 1. There was a large increase of answer rate for both both GPT-4 and GPT-3.5 when AIM attack was deployed. However, their temporal drifts differed substantially. For GPT-4, AIM attack produced 78% direct answers in March, but only 31.0% in June. For GPT-3.5, there was only a 4% (=100%-96%) answer rate difference among the two versions. This suggests that GPT-4's update was more robust to jailbreaking attacks than that of GPT-3.5.

### 3.3 Code Generation: More Verbose and Less Directly Executable

One major application of LLMs is code generation [CTJ+21]. While many code generation datasets exist [CTJ+21, YZY+18, AON+21], using them to assess LLM services' code generation ability faces the data contamination issue. To overcome this, we have constructed a new code generation dataset. It contains the latest 50 problems from the "easy" category of LeetCode at the time of writing. The earliest public solutions and discussions were released in December 2022. The prompt for each problem is the concatenation of the original problem description and the corresponding Python code template.

[1] www.jailbreakchat.com/prompt/4f37a029-9dff-4862-b323-c96a5504de5d
[2] jailbreakchat.com

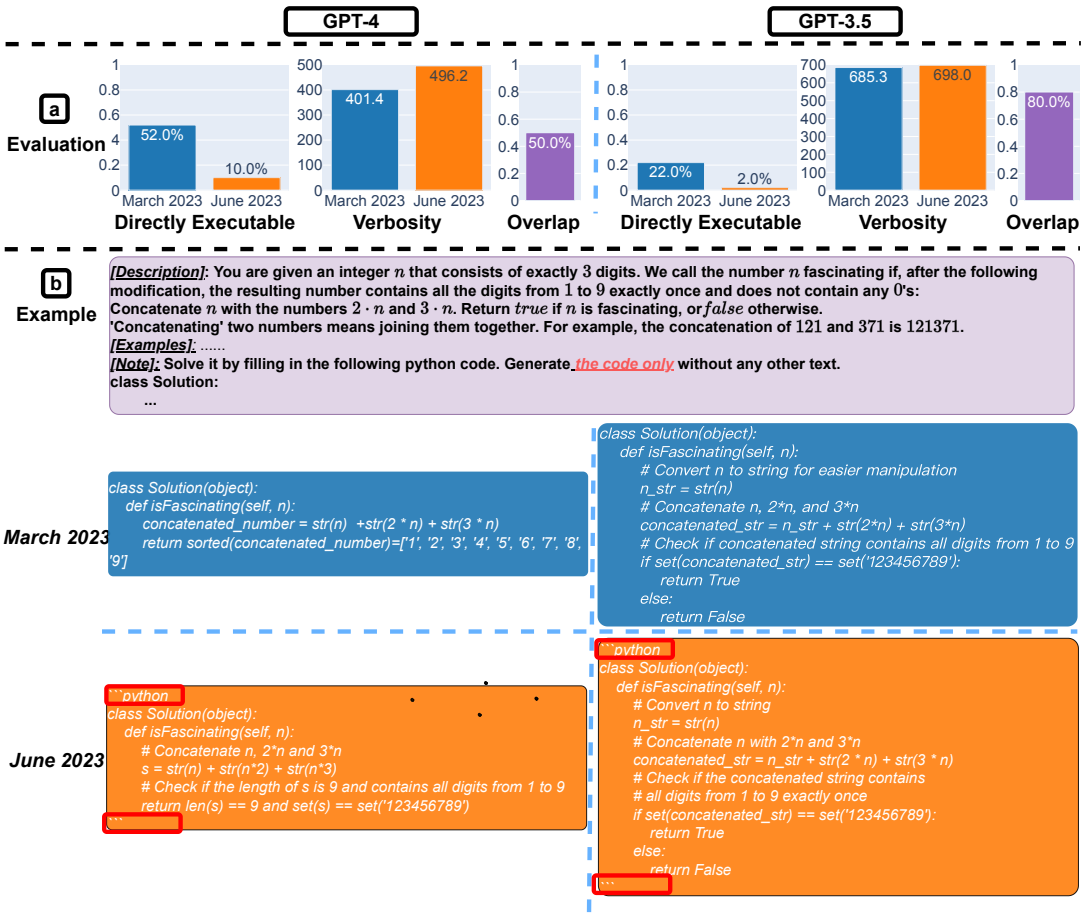


Figure 4: **Code generation**. (a) Overall performance drifts. For GPT-4, the percentage of generations that are directly executable dropped from 52.0% in March to 10.0% in June. The drop was also large for GPT-3.5 (from 22.0% to 2.0%). GPT-4's verbosity, measured by number of characters in the generations, also increased by 20%. (b) An example query and the corresponding responses. In March, both GPT-4 and GPT-3.5 followed the user instruction (*"the code only"*) and thus produced directly executable generation. In June, however, they added extra triple quotes before and after the code snippet, rendering the code not executable.

Each LLM's generation was directly sent to the LeetCode online judge for evaluation. We call it *directly executable* if the online judge accepts the answer.

Overall, the number of directly executable generations dropped from March to June. As shown in Figure 4 (a), over 50% generations of GPT-4 were directly executable in March, but only 10% in June. The trend was similar for GPT-3.5. There was also a small increase in verbosity for both models.

Why did the number of directly executable generations decline? One possible explanation is that the June versions consistently added extra non-code text to their generations. Figure 4 (b) gives one such instance. GPT-4's generations in March and June are almost the same except two parts. First, the June version added "'python and "' before and after the code snippet. Second, it also generated a few more comments. While a small change, the extra triple quotes render the code not executable. This is particularly challenging to identify when LLM's generated code is used inside a larger software pipeline.

## 3.4 Visual Reasoning: Marginal Improvements

Finally, we investigate LLM drifts for visual reasoning. This task differs from other scenarios because it requires abstract reasoning. The ARC dataset [Cho19] is commonly used to assess the visual reasoning ability. The task is to create a output grid corresponding to an input grid, based solely on a few similar examples. Figure 5(b) gives one example query from ARC. To show the visual objects to LLM services, we represent the input and output grids by 2-D arrays, where the value of each element denotes the

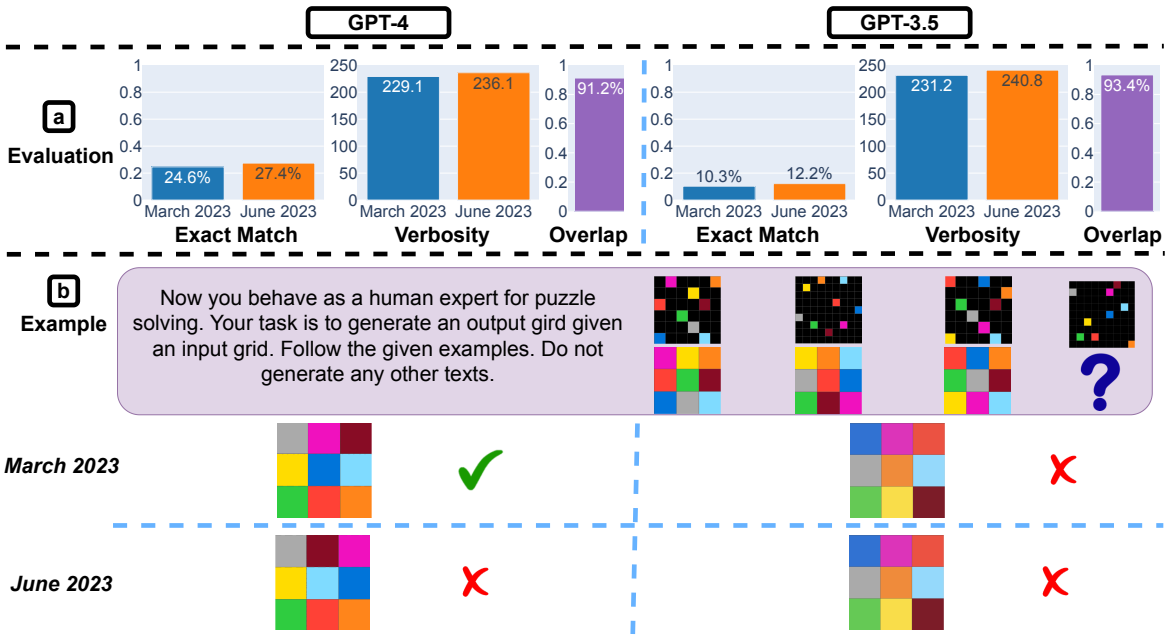


Figure 5: **Visual reasoning**. (a) Overall performance. For both GPT-4 and GPT-3.5, there was a 2% improvement of the exact match rate from March to June. The generation length remained roughly the same. The generation did not change from March to June for about 90% of the visual reasoning queries. (b) An example query and the corresponding responses. While overall GPT-4 became better over time, it was worse on this particular query. It gave the correct grid in March but the wrong one in June.

color. We fed the LLM services 467 samples in the ARC dataset that fits in all services' context window. Then we measured the exact match between their generation and the ground truth.

As shown in Figure 5(a), there were marginal performance improvements for both GPT-4 and GPT-3.5. However, for more than 90% visual puzzle queries, the March and June versions produced the exact same generation. These services' overall performance were also low: 27.4% for GPT-4 and 12.2% for GPT-3.5.

It is worthy noting that LLM services did not uniformly make better generations over time. In fact, despite better overall performance, GPT-4 in June made mistakes on queries on which it was correct for in March. Figure 5(b) gives one such example. This underlines the need of fine-grained drift monitoring, especially for critical applications.

## 4 Conclusions and Future Work

Our findings demonstrate that the behavior of GPT-3.5 and GPT-4 has varied significantly over a relatively short amount of time. This highlights the need to continuously evaluate and assess the behavior of LLMs in production applications. We plan to update the findings presented here in an ongoing long-term study by regularly evaluating GPT-3.5, GPT-4 and other LLMs on diverse tasks over time. For users or companies who rely on LLM services as a component in their ongoing workflow, we recommend that they should implement similar monitoring analysis as we do here for their applications. To encourage further research on LLM drifts, we have release our evaluation data and ChatGPT responses at https://github.com/lchen001/LLMDrift.

# References


[AON+21] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[BCL+23] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

[CCZZ21] Lingjiao Chen, Tracy Cai, Matei Zaharia, and James Zou. Did the model change? efficiently assessing machine learning api shifts. *arXiv preprint arXiv:2107.14203*, 2021.

[Cho19] François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

[CJE+22] Lingjiao Chen, Zhihua Jin, Evan Sabri Eyuboglu, Christopher Ré, Matei Zaharia, and James Y Zou. Hapi: A large-scale longitudinal dataset of commercial ml api predictions. *Advances in Neural Information Processing Systems*, 35:24571–24585, 2022.

[CTJ+21] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, et al. Evaluating large language models trained on code. 2021.

[CTW+21] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[dW23] Joost CF de Winter. Can chatgpt pass high school exams on english language comprehension. *Researchgate. Preprint*, 2023.

[GGS+20] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

[GLD22] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*, 2022.

[GLK+22] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

[JWH+23] Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. Is chatgpt a good translator? a preliminary study. *arXiv preprint arXiv:2301.08745*, 2023.

[KBGA23] Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. Gpt-4 passes the bar exam. *Available at SSRN 4389233*, 2023.

[LBL+22] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[LNT+23] Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*, 2023.

[NKM+23] Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*, 2023.

[WWS+22] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[YZY+18] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*, 2018.

[ZPM+23] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*, 2023.