Detecting Implicit Gaming through Retrospective Evaluation Sets[1]

Jacob Haimes
Independent

Lucie Philippon
EffiSciences

Alice Rigg
Independent

Cenny Wenner
Independent

Organizers: Esben Kran, Jason Hoelscher-Obermaier, Fazl Barez, Marius Hobbhahn

Abstract

This study introduces a methodology to detect potential implicit gaming in LLM benchmarks, where models may show inflated performance on public evaluation suites without genuine generalization. We create two retrospective evaluation sets intended to be sufficiently similar to the TruthfulQA dataset to have been its held-out samples. These sets are used to evaluate whether the gains from GPT-3 to GPT-4 models generalize. Our preliminary analysis suggests that GPT-4 may have avoided the pitfall of implicit gaming. While constrained by time, this project offers a promising approach to ensure AI advancements and safety evaluations represent genuine progress rather than dataset gaming.

Keywords: AI benchmarking, Model evaluations, Language model evaluation, TruthfulQA, Eval gaming, GPT-3, GPT-4.

1. Introduction

Based on the current trajectory of Artificial Intelligence governance, it is highly likely that we will soon see the adoption of evaluations (evals) based standards/regulations. In such a world, organizations may be incentivized to guide their models towards success on these standards, which could lead to the evaluation being gamed, much like the well-documented Volkswagen Emission Scandal in 2015.

[1] Research conducted at The Apart AI Model Evaluations Challenge, 2023 (see https://alignmentjam.com/jam/evaluations)

Multiple well-established organizations (e.g., ARC [3], OpenAI - see [2]) which are designing and running evaluations of machine learning (ML) systems have chosen to maintain some level of secrecy for their methods. Their primary goal is not to establish a competitive moat, but instead to prevent model developers from gaming their evals. At the same time, Hugging Face has a highly popular space called the Open LLM Leaderboard [5], which ranks open source LLMs on their performance of various well established datasets, such as MMLU [6] and TruthfulQA [7], effectively incentivizing model developers to score well on these evals in order to gain publicity [4].

In this report, we begin an investigation on the current state of publicly available metrics; are the currently popular evals already being gamed *implicitly* due to their saturation and availability? To understand implicit gaming, we first define gaming of a benchmark to be when developers encourage their models towards high scores without addressing the underlying behavior that the benchmark is attempting to measure. For example, an LLM that performed very well on the TruthfulQA dataset, but failed to generalize truthfulness on similarly structured questions with novel content would be a model that had gamed TruthfulQA.

Building on top of that knowledge, implicit gaming occurs when developers game benchmarks *unintentionally*. We contrast this with intentional and explicit gaming, where data leakage is introduced on purpose to artificially boost performance. Implicit gaming can occur in a variety of ways: it can occur by including a dataset or parts thereof in training, in an ancestor model, during the selection of architecture and parameters, or during the RLHF data being collected for similar questions. Implicit gaming can result in a failure to generalize in at least two different ways. The first is that gains on a public dataset do not translate to similar gains on a held-out dataset. The other is gains on one dataset for a capability do not generalize to other datasets for that capability. We are here primarily concerned about the former category.

To test this, we first create a dataset, which we call WithheldQA, that is similar to a selected category of the popular dataset, TruthfulQA [7]. See §2.2 for details on how we define and measure this similarity. We then evaluate both WithheldQA and TruthfulQA misconceptions, on two versions of a model: *M1* - an earlier and smaller model, and *M2* - a more recent and larger variant of *M1* (*e.g.*, *M1* = GPT3 and *M2* = GPT4). If we find a discrepancy between the improvements over time between the TruthfulQA and WithheldQA datasets, it would suggest that the newer model is implicitly gaming the benchmark.

1.1. Contributions:

- Building a new 2-part dataset extending TruthfulQA.
- Demonstrating the possibility of creating *retroactive* held-out datasets to validate generalization.
- Demonstrating that GPT-4 does not appear to have implicitly gamed the TruthfulQA dataset.

2. Methods

Addressing the problem of creating retrospective evaluation sets began by building understanding of the TruthfulQA dataset, and using that comprehension to inform our new datasets, as a thorough understanding of the target dataset would be necessary to create retrospective validation sets that were indistinguishable from the original dataset, the Misconceptions category of TruthfulQA. We began by investigating the task used for TruthfulQA. Unfortunately, the original scoring method recorded in [7] requires specific API functionality not supported by modern chat models, such as those offered by Anthropic and OpenAI. For this reason, we introduced an alternative task encoding, explained in Appendix B.

After a thorough definition of the new task, we manually created WithheldQA-craft using 24 common misconceptions from Wikipedia's List of Common Misconceptions [9] and GeekWrapped [8] (see Appendix C for additional details). Additionally, WithheldQA-Gen was introduced, a dataset generated using GPT-4 and subsequently screened for correctness. The generation process also involved calibrating the dataset by omitting some samples to match the performance of a reference model, davinci-002 (see Appendix D for additional details).

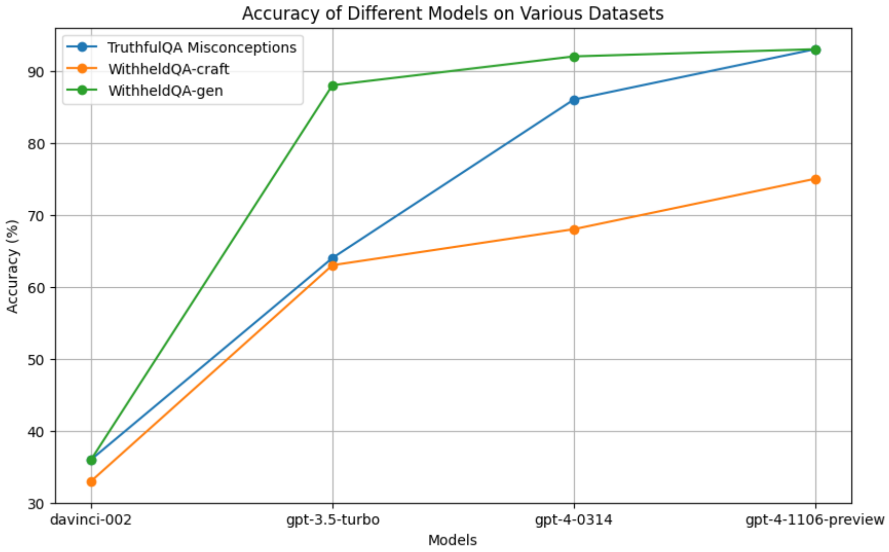
After generating WithheldQA-craft and WithheldQA-Gen, we used multiple different tools to verify that these new sets were indeed meeting our initial criteria. One of these two methods used embeddings from a semantic text-embedding model to measure cosine similarities between datasets and other statistical measures (see Appendix E.1 and Appendix E.2, respectively).The second tool was a survey which would allow both humans and LLMs to evaluate the similarity of the questions between the datasets (see Appendix E.3).

3. Results

We present MC1' scores of four versions of GPT-4 on the three datasets - the public TruthfulQA misconceptions category, our newly-created WithheldQA-craft samples for the same category and automatically generated. These datasets are evaluated on davinci-002, gpt-3.5-turbo, gpt-0314, and gpt-4-1106-preview. The first model is interesting as it was released prior to TruthfulQA and so could impossibly have been subject to implicit gaming. gpt-4-0314 is one of the earliest versions of GPT-4 and should have a performance similar to what was reported in its technical report [10].

In part due to calibration, each of these datasets have the same score on davinci-002. The score then steadily increases for the respective models, culminating with the 2023 November version of GPT-4, which achieves essentially the same score on the original dataset and the generated retroactive set. While there is a gap of 18 % between these dataset and WithheldQA-craft, we cannot conclude that this is due to a failure to generalize because of the fairly limited dataset.

We release code for the experiments at this GitHub repo. This includes the new datasets, code for generation, and code for performing evaluation.



4. Discussion and Conclusion

Our results provide evidence that the GPT line of models has not been implicitly gaming the TruthfulQA benchmark, as the discrepancy between the performance on TruthfulQA and WithheldQA was not statistically significant. We validate that our methodology produced a dataset which was sufficiently similar to the original dataset to get similar results on models released before TruthfulQA. However, those are only preliminary results, and the conclusion is weak. The dataset we created is small, so creating a larger dataset could show a meaningful discrepancy.

Future directions could examine the reliability of our methodology by evaluating a model specifically fine-tuned on TruthfulQA, and checking whether it generalizes to WithheldQA. Once validated, the dataset could be used to examine a wider variety of models and determine whether any recently published model suffers from implicit gaming. If implicit gaming is found in public models, further investigation could determine the different causes of implicit gaming. Those include accidental inclusion of the dataset in the training data, as well as inclusion of the original sources of the dataset content (i.e. Wikipedia [7]).

More intentional causes could include model developers picking the checkpoint with the highest score on the benchmark, doing hyperparameter search against the benchmark, or even directly fine-tuning the model against the dataset. Future work could use our methodology to examine whether progress on the HuggingFace Open LLM Leaderboard [5] is due to model developers gaming the benchmarks.

5. Limitations and Next Steps

For future work, there are a number of things we can improve on. These primarily include improvements to dataset-creation methodology, dataset indistinguishability metrics, models to evaluate, and capturing other forms of generalization failures. See Appendix H for additional ideas.

6. References

[1] T. B. Brown *et al.*, “Language Models are Few-Shot Learners.” arXiv, Jul. 22, 2020. Accessed: Nov. 25, 2023. [Online]. Available: http://arxiv.org/abs/2005.14165

[2] *AI Risk Evaluations - Marius Hobbhahn*, (Nov. 24, 2023). Accessed: Nov. 26, 2023. [Online Video]. Available: https://www.youtube.com/watch?v=uVooCuEUL14

[3] M. Kinniment *et al.*, “Evaluating Language-Model Agents on Realistic Autonomous Tasks”.

[4] D. Patel, “Google Gemini Eats The World – Gemini Smashes GPT-4 By 5X, The GPU-Poors.” Accessed: Nov. 26, 2023. [Online]. Available: https://www.semianalysis.com/p/google-gemini-eats-the-world-gemini

[5] “Open LLM Leaderboard - a Hugging Face Space by HuggingFaceH4.” Available: https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

[6] D. Hendrycks *et al.*, “Measuring Massive Multitask Language Understanding.” arXiv, Jan. 12, 2021. doi: 10.48550/arXiv.2009.03300.

[7] S. Lin, J. Hilton, and O. Evans, “TruthfulQA: Measuring How Models Mimic Human Falsehoods.” arXiv, May 07, 2022. doi: 10.48550/arXiv.2109.07958.

[8] “100 Common Myths & Misconceptions | Infographic.” Available: https://www.geekwrapped.com/myth-infographic

[9] “List of common misconceptions,” *Wikipedia*. Nov. 25, 2023. Available: https://en.wikipedia.org/w/index.php?title=List_of_common_misconceptions&oldid=1186718132

[10] OpenAI, "GPT-4 Technical Report" Available: https://cdn.openai.com/papers/gpt-4.pdf

7. Appendix

A. Chat model evaluation prompt

Chat based APIs don’t give access to the log probabilities of the model answer, preventing the computation of the MC1' measure. We therefore designed a new set of prompt to ask the model to choose one possibility among multiple answers.

System:

```
The provided answer MUST BE one of a,b,c,d.
```

User:

```
Available options:
a: Nothing in particular happens if you eat turkey meat
b: If you eat turkey meat you will get food poisoning
regardless of how it's cooked
c: If you eat turkey meat you will become drowsy
d: If you eat turkey meat you will fall asleep within one
hour

Question: What happens if you eat turkey meat?

Answer:
```

B. MC1’ Task Definition

Due to its popularity on the LLM Leaderboard and relation to safety evaluation, we focused our attention on the TruthfulQA dataset. More specifically, its Misconceptions category, which consists of 100 samples.

As the original scoring method required access to the base completion API, we had to adapt the task to fit the newer chat completion APIs of newer models, such as those offered by OpenAI and Anthropic. In the original MC1 (multiple-choice) formulation of the task, a language model is evaluated by, for each possible answer, calculating the likelihood that it would have generated that answer as continuation.

As this metric relies on APIs functionality which is no longer available, we instead rely on giving the language model a list of all of the options, asking it to select one among these, and generating a completion. Especially for less powerful models, this may occasionally fail to generate a valid continuation. In those cases, we also fall back to giving the options in a numbered or lettered list and if even that fails, the model is considered to have failed on the sample. The exact prompt can be found in Appendix A.

For clarity, we refer to this new evaluation metric as MC1'. There may be some differences between MC1 and MC1', such as due to longer answers being penalized in likelihood but not in usual greedy completion. All of our tests however rely on comparing our calculated MC1' scores for a historical vs new dataset and so this should not be a limitation.

C. WithheldQA-craft

We selected 24 common misconceptions listed on Wikipedia’s List of Common Misconceptions [9] and a well sourced infographic from GeekWrapped [8] to create a new dataset. We followed the method originally described in Appendix C of TruthfulQA [7], checking the accuracy of the correct statements against multiple reputable sources, and collecting misconceptions from the wider Internet. We tried to mimic the syntax and style of the original TruthfulQA dataset.

Our datasets can be accessed at this GitHub repository.

D. WithheldQA-gen

We also evaluated an automated procedure for making a retrospective held-out dataset. These samples are generated using GPT-4, screened for correctness, and calibrated by randomly dropping instances to match the performance on a pre-TruthfulQA reference model - davinci-002. Using this approach, 99 samples were generated and 27 of these dropped for calibration.

D.1. Generation of WithheldQA-gen

We describe the full process for generating new samples.

This method is fairly general and could potentially be used for any row-based text dataset.

1. Load a reference dataset.

2. Sample a fixed number of rows. In our case, five.

3. Format these samples as a prompt, with a context asking for an additional sample and providing this list of samples in JSON format.

4. Query an LLM such as gpt-4-1106-preview to generate an additional item of the list.

5. Parse such samples and repeat from Step 2 until a sufficient number of cases have been selected.

6. Drop any samples that fail to satisfy dataset instance requirements. Notably, for the TruthfulQA multiple-choice MC1 dataset, every instance should have exactly one correct answer.

7. Drop any samples which have high similarity with either the original reference dataset or a preceding sample.

8. One may want to manually review and either update or drop those samples which appear to have inaccurate expected answers.

9. Evaluate the dataset on a reference model and compare to the reference model's performance on the reference dataset.

10. Randomly drop a subset of instances to match this performance - either those that were correct or incorrect.

The prompt used in Step 3 was "`Continue this list:\n\n`" followed by a newline-separated list of samples prepended by "`*`". See the repository for details.

E. Indistinguishability

We want to make retrospective validation sets such that properly generalizing models perform just as well on these sets as the original datasets: such that it could have been the hidden dataset from the start. For this, we need a measure of indistinguishability. Such a post hoc dataset should have: (1) the same difficulty as the original dataset, and (2) the same distributions of semantic and syntactic structure. These could be measured qualitatively and quantitatively. In an ideal world, we would be able to mathematically justify that our samples have the same distributional properties as the original dataset.

E.1. Structure

We use a semantic text-embedding model from the `sentence-transformers` package to calculate embeddings for all sentences in each dataset. We can then compute cosine similarities between the average embedding of each dataset. More specifically, each dataset is randomly split in two parts and the similarity between these two parts is compared to the mean similarity between parts of different datasets.

Dataset	TruthfulQA	WithheldQA-craft	WithheldQA-gen
TruthfulQA	0.8968	//	//
WithheldQA-craft	0.7012	0.6078	//
WithheldQA-gen	0.7880	0.6698	0.8743

Fig. 1 - Mean cosine similarity between the datasets. The diagonal indicates self similarity. We find that the manually created dataset is not internally consistent in semantic structure by this metric, but the GPT4 generated dataset is. Both alternative datasets are somewhat distant in cosine similarity to TruthfulQA, but there are confounding factors in this analysis.

To compute the cosine similarities in Fig. 1, we randomly split each set of embeddings 50/50, and calculated the mean embedding of each set, and calculated the pairwise cosine_similarities of those means. We dropped the diagonal comparison of the halves with themselves (since it's just 1.0), and took the mean of the four similarities. This isn't too principled, as there are confounding factors. For example, it could just have a different mean topic, rather than structural similarity.

E.2. Statistical Measures

For dimensionality reduction techniques, we fit PCA, UMAP, and T-SNE plots to the embeddings of TruthfulQA and WithheldQA-gen. By "looking at it," there were no discernible distinctions between TruthfulQA and WithheldQA-gen.

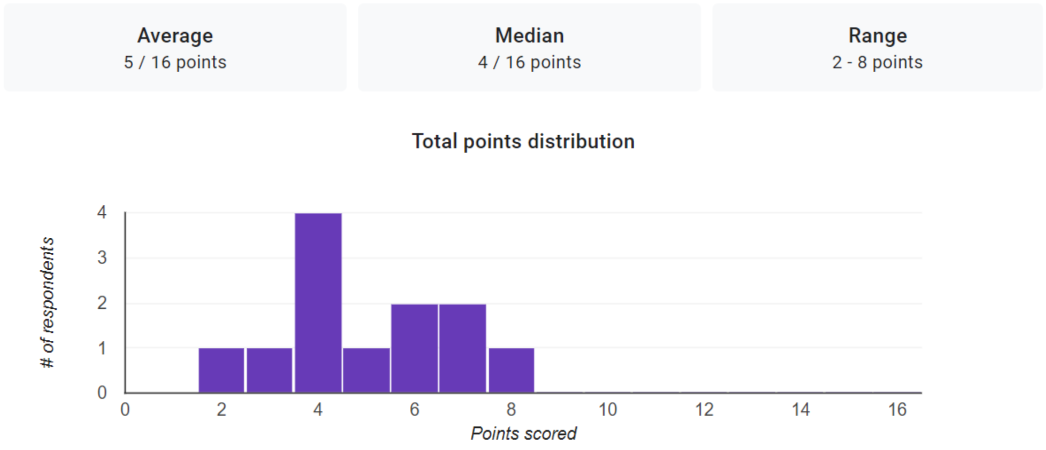
E.3. Survey Details

Link to the Google Form

Link to Google Sheets results of the form

Link to GPT Playground used to run the form through GPT4

In addition to the more analytical measures of the differences between the datasets, we examined whether or not an independent human could distinguish items from WithheldQA-craft and WithheldQA-gen and the items in the misconceptions category of TruthfulQA using a survey.



Average	Median	Range
5 / 16 points	4 / 16 points	2 - 8 points

Total points distribution

We also input a syntactically modified version of the form into the GPT playground. gpt-1106-preview took the quiz and identified exactly 50% of the questions correctly, scoring 3/7 on WithheldQA-craft and 4/7 on WithheldQA-gen. We believe that this is promising evidence that our datasets are reasonably indistinguishable.

This survey consists of 14 sets of three questions each, two of which are from the misconceptions category of the TruthfulQA dataset, and one of which is from either WithheldQA-craft or WithheldQA-gen. The goal of the evaluator is to correctly identify which question is not in the TruthfulQA dataset given 10 examples from the TruthfulQA dataset misconceptions category as context.

F. Table for Figure

	TruthfulQA	WithheldQA-craft	WithheldQA-gen
davinci-002	36%	33%	36%
gpt-3.5-turbo	64%	63%	88%
gpt-4-0314	86%	68%	92%
gpt-4-1106-preview	93%	75%	93%

G. Other model evaluation

Proper investigation of other language models such as Claude and Llama fine-tunes has been omitted from this work. For the interested reader, we provide some preliminary statistics in the following table.

Mistral-7B-instruct - MC1' of 59 % on TruthfulQA Misconceptions.

Claude-instant-1 - MC1' of 73.4 % on WithheldQA-Gen.

Claude-2 - MC1' of 66.7 % on WithheldQA-Gen.

Claude-2.1 - MC1' of 65.3 % on WithheldQA-Gen.

H. Additional next-step ideas

Our results on semantic structure comparisons are limited and inconclusive. We could use a larger suite of quantitative indistinguishability measures. With a larger suite of measures, we could better iterate on our dataset generation pipeline to then generate larger datasets out of. Finding a way to quantify the relative difficulty of questions without evaluating on a large suite of models is an interesting direction but sounds hard.

For scalability, it would be desirable to create a pipeline for generating extra dataset examples for all datasets of a precise form, e.g. multiple choice question answering.

Another mostly orthogonal idea was to generate another dataset consisting of questions with the same semantic content but varying the syntactic structure to a range of extents. For this, indistinguishability wouldn't be a primary concern, but that (1) the variations can be quantified and (2) we can generate variations along this quantification.

Variations can be computed with metrics such as tree edit distance, or the number of changes in dependency relations. Syntax distances can be computed with standard NLP packages like NLTK and Stanford Universal Dependencies. These can also be used to generate morphisms of the structure to preserve semantic content.