BENCHMARK INFLATION: REVEALING LLM PERFORMANCE GAPS USING RETRO-HOLDOUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

The training data for many Large Language Models (LLMs) is contaminated with test data. This means that public benchmarks used to assess LLMs are compromised, suggesting a performance gap between benchmark scores and actual capabilities. Ideally, a private holdout set could be used to accurately verify scores. Unfortunately, such datasets do not exist for most benchmarks, and post-hoc construction of sufficiently similar datasets is non-trivial. To address these issues, we introduce a systematic methodology for (i) retrospectively constructing a holdout dataset for a target dataset, (ii) demonstrating the statistical indistinguishability of this *retro-holdout* dataset, and (iii) comparing LLMs on the two datasets to quantify the performance gap due to the dataset's public availability. Applying these methods to TruthfulQA, we construct and release Retro-Misconceptions, on which we evaluate twenty LLMs and find that some have inflated scores by as much as 16 percentage points. Our results demonstrate that public benchmark scores do not always accurately assess model properties, and underscore the importance of improved data practices in the field.

> "The enemy of truth is blind acceptance."
> –Anonymous
>
> *Lin et al., 2022*

1 INTRODUCTION

Many have begun to question the reliability of public benchmarks in assessing large language models (Alzahrani et al., 2024; Zheng et al., 2024; Fourrier et al., 2023). Discrepancies between benchmark scores and practical capabilities raise concern (Li et al., 2024b), and strong incentives for higher scores (Fourrier et al., 2024) suggest that optimizing benchmark performance could take precedence over real-world effectiveness and safety. This phenomenon, akin to specification gaming (Krakovna et al., 2020), is termed *evaluation gaming* – processes leading to a systematic gap between benchmark performance and practical utility.

Extensive evidence of evaluation data being included in training data (Sainz et al., 2024; Oren et al., 2023; Schaeffer, 2023; Shi et al., 2023; Jiang et al., 2024; SLAM-group, 2023) suggests that evaluation gaming is occurring. However, proving the existence of a statistically significant performance gap between a specific evaluation task and an analogous real-world task would require access to an independently and identically distributed (IID) split of the benchmark which we know could not have had an impact on any aspect of model development.

This is the idea of *holdout* datasets, which are used to assess a machine learning model's unbiased performance after training. By definition, a holdout dataset comes from the same distribution as its corresponding target dataset, meaning that any evaluation conducted on both datasets should have the same result within some statistical tolerance (James et al., 2023). Importantly, holdout datasets also are kept hidden during the development process. Together, these two properties imply that comparing a model's performance on a public benchmark and a corresponding holdout dataset could reveal whether the public benchmark has influenced any aspect of the design, training, or validation process. Unfortunately, holdout datasets for benchmarks are typically not available; benchmark developers usually release all evaluation data, although there are notable exceptions, e.g. Li et al. (2024a).

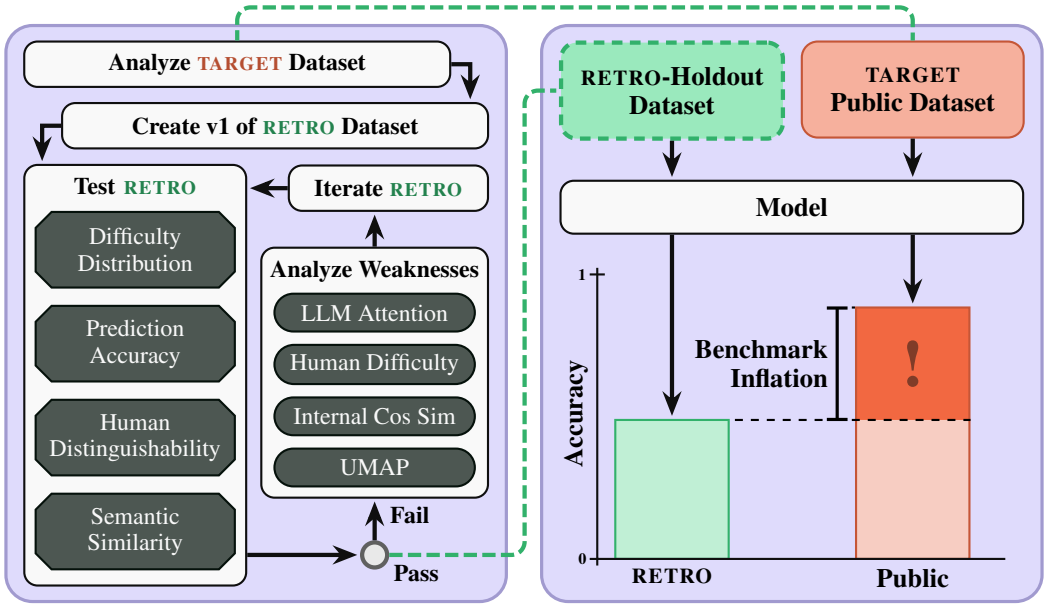


Figure 1: Visualization of our methodology. The left panel summarizes the process for constructing a retro-holdout dataset, while the right panel illustrates how to leverage such a dataset to quantify benchmark inflation.

To resolve this, we propose *retroactive holdout*, or *retro-holdout*, datasets, which are verified to be sufficiently similar to their corresponding target dataset through various tests, despite being created independently and retroactively. Utilizing a retro-holdout, we can quantify the evaluation performance gap of any given model. We detail our methodology for creating and validating retro-holdout datasets, along with multiple recommendations and tools for generating such datasets. Using the TruthfulQA[1] evaluation (Lin et al., 2022), we conduct a case study to quantify performance gaps for twenty contemporary models. Our results conclusively indicate that evaluation gaming *is* occurring, underscoring the need for improved data practices in the domain.

1.1 Contributions

In this work, we:

- Present a novel process for constructing retro-holdout datasets.
- Release Retro-Misconceptions, a retro-holdout dataset for TruthfulQA, which can be used to quantify the performance gaps of a model on the original dataset.[2]
- Evaluate twenty models using Retro-Misconceptions to demonstrate measurable score inflation.

2 Methods

2.1 Understanding the retro Holdout Framework

Holdout datasets were first used in machine learning to accurately assess model performance on a given task. A holdout set is a randomly selected subset of the same set of observations as the training dataset, and is strictly excluded from the development process (James et al., 2023). Unlike conventional holdout sets, retro-holdout datasets are created after the initial release of a dataset,

[1]TruthfulQA was chosen due to its safety relevance and widespread use (Zhao et al., 2023; Naveed et al., 2024; Bai et al., 2023; Cui et al., 2024; Fourrier et al., 2024).

[2]Retro-Misconceptions is only guaranteed to be accurate on models with a training cutoff date prior to January 1st, 2024, since that as that is when portions of the new dataset became available on the web.

meaning we cannot assume they have the same properties that a hypothetical holdout set would have. We refer to Appendix A for a formalization of this claim.

In brief, we rely on a standard assumption in machine learning that the public and post-hoc retro-holdout datasets consist of independent samples from two possibly different distributions Hastie et al. (2009); Shalev-Shwartz & Ben-David (2014). To establish that the retro-holdout can be used as a holdout set for the public benchmark, we must show that both datasets could have been sampled from the same distribution. We construct four statistical tests, one permutation test and three binomial tests, to reject the hypothesis that two sets were sampled from the same distribution. A proposed dataset cannot be considered a retro-holdout for a given public benchmark unless all four tests fail to reject the hypothesis of a shared distribution.

This verification process sets our methodology apart from a more standard dataset extension, as it mandates that our retro-holdout is an assessment of exactly the same task as the original benchmark, making the only difference between the two the variable we care about: public availability of a dataset. The lengths we have gone in order to reach this level of rigor are extensive; the retro-holdout framework does not make use of LLMs for any aspect of dataset creation. This substantially increases the time cost of our process, but ensures that language models do not bias our results.

Our method is designed for labeled datasets, which have inputs and expected outputs for models. We note that out of distribution (OOD) testing has sometimes been incorrectly characterized as using holdout datasets. In this work, we use the original definition of holdout sets; evaluation sets constructed to probe OOD performance are not considered.

For brevity, we define

$$\text{TARGET} := \text{an arbitrary, publicly available benchmark,}$$
$$\text{RETRO} := \text{a retro-holdout dataset for TARGET.}$$

2.2 Creating a RETRO

Initially, the methodology used for crafting a RETRO should be heavily informed by the original process used to create the TARGET. To promote similarity, we recommend using a representative entry, randomly drawn from TARGET, without replacement, as a basis for creating each new entry in RETRO. We include a short guide for this initial creation in Appendix D.

2.2.1 Sufficient Indistinguishability

Establishing with absolute certainty that the two datasets have originated from the same distribution is impossible. Therefore, we resort to multiple statistical tests designed to test the null hypothesis that TARGET and RETRO have a common origin. If the result of each test indicates that we cannot reject our null hypothesis, we designate our RETRO to be sufficiently indistinguishable from TARGET. While it is theoretically possible to construct any number of tests to evaluate the similarity between two datasets, practical considerations guide us to four key tests that provide a thorough assessment:

- **Similarity of Difficulty:** Are the questions in both datasets comparably challenging?
- **Semantic Embedding Similarity:** What is the likelihood that a distribution of cosine similarities between sentence embeddings similar to that of RETRO have been pulled from the same distribution as TARGET?
- **Prediction Accuracy:** Can a model, fine-tuned on randomized splits of the datasets, differentiate between TARGET and RETRO?
- **Human Distinguishability:** Can humans differentiate between TARGET and RETRO?

We designate the two datasets as *sufficiently indistinguishable* if all four tests fail to reject the null hypothesis at a p-value of 5%.

Similarity of Difficulty To verify that the two datasets have comparable difficulties, we use both to evaluate models with a training cutoff date prior to the release of the TARGET, or *pre-release* models. These models could not have been affected by the TARGET, as it had not yet been released; model performance on both datasets should be comparable, with a margin of statistical uncertainty.

This reduces to a two-proportion binomial test with the null hypothesis of equal success probability. For further information on the evaluation task, refer to §2.3.

We note that with access to many LLMs of varying capability levels, this test combined with simple human assessment would likely suffice to determine sufficient indistinguishability between the two datasets. However, performance of cutting-edge models continues to improve, meaning that pre-release models are practically guaranteed to be less capable than contemporary models, assuming they are accessible at all. These constraints are expanded on on Appendix G. To address this limitation, we use a number of techniques to amplify pre-release performance: allowing the model to choose multiple answers (top-k), including examples of other questions within the dataset (5-shot), and using the 'helpful' prompt from Lin et al. (2022).

Prediction Accuracy We adopt a modification of prediction accuracy as defined by Dankar & Ibrahim (2021) to determine if a machine learning model can differentiate between the datasets. Contrary to the conventional use of logistic regression in synthetic data evaluations (Dankar & Ibrahim, 2021), we fine-tune BERT (Devlin et al., 2019) on a prediction task. This choice was informed by BERT's capability to capture nuanced semantic relationships within text, which are crucial for accurately assessing the subtle distinctions or similarities between dataset entries. If the model's prediction accuracy is approximately the same as random performance, 50%, we can conclude that the model cannot differentiate between the two datasets. Under a null hypothesis, this simplifies to a binomial test with success probability of $1/2$.

To calculate the prediction accuracy score, each dataset is split into five folds. One split from each dataset is withheld, and BERT is fine-tuned to accurately label an entry as either RETRO or TARGET on the remaining data. The model's prediction accuracy on the holdout splits is measured, and then the process is repeated such that each split is used for testing.

Semantic Embedding Similarity We perform a random permutation test (Fisher, 1974; normaldeviate, 2012; Hemerik, 2024) using semantic embeddings from Sentence Transformers (Reimers & Gurevych, 2019). We define the test statistic as the mean of all pairwise cosine similarities between embeddings. To obtain a sufficiently tight bound, we use a sample size of $N = 10000$. This test is formally defined in Appendix A.

Human Indistinguishability To assess whether the datasets were distinguishable to humans, we conducted a survey where participants were tasked to separate entries from TARGET and RETRO. Participants were shown ten labeled entries from each dataset for contextual understanding, followed by a series of ten tests, each comprising of three dataset entries – two from TARGET and one from RETRO. All entries are drawn without replacement to ensure unique samples throughout the survey. Under the null hypothesis, this is a binomial test with success probability $1/3$.

We also implement a variation of this test using GPT-4o as the evaluator to compare human and model performance. See Appendix F for comprehensive details on the survey methodology, including specifics on participant recruitment, the structure of the test, and survey instructions.

2.2.2 An Iterative Process

Creating a RETRO that meets our rigorous standards for sufficient indistinguishability is non-trivial and will likely require iteration. Acknowledging this, and considering the time-intensive nature of dataset generation, efficiency is quite important. We have created a number of tools that aid in high-level iteration:

- **Fine-Tuned Prediction Model Attention:** A BERT model (Devlin et al., 2019) is fine-tuned to classify entries as belonging to either TARGET or RETRO. *Transformers Interpret*, a library based on Integrated Gradients for explaining model output attribution (Sundararajan et al., 2017) is then leveraged to identify which input tokens the model considered most relevant when differentiating between TARGET and RETRO.
- **Datapoint Embeddings:** `all-mpnet-base-v2` is used through the HuggingFace *Sentence Transformers* library, to create vector representations for all data points (Reimers & Gurevych, 2019). These embeddings are then taken as the basis for the following three

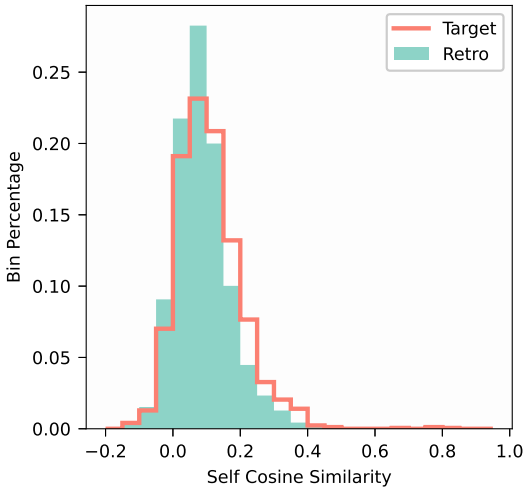


Figure 2: Example output from the Internal Cosine Similarity Distribution tool. This specific plot indicates that entries within the TARGET were systematically more similar by a small amount, which led the team to further scrutinize word frequencies.

tools; when analyzed in conjunction they can provide meaningful insights on general similarity trends, outlier detection, and topic clustering.

- **Embedding Space Visualization:** We employ Uniform Manifold Approximation and Projection (UMAP) to project these embedding vectors onto a two-dimensional plane (McInnes et al., 2018). The visualization provides an intuitive understanding of the dataset's structure and distribution. An example output of this visualization tool is provided in Figure 6.
- **Internal Cosine Similarity Distribution:** To assess similarity between entries within the datasets we plot histograms of pairwise cosine similarities of datapoint embeddings. This representation aids in identifying outliers and assessing overall similarity within the datasets, as demonstrated in Figure 2.
- **Largest Internal Cosine Similarity Comparison:** We highlight the ten entry pairs with the highest cosine similarities in both datasets, providing a direct comparison of the most similar entries and their respective values.

Still, it is quite possible that insights found with these tools will not be enough to ensure sufficient indistinguishability. Additional documentation for using the tools, as well as recommendations for this process, are detailed in Appendix E.

2.3 Evaluating Models

TruthfulQA Lin et al. (2022) was designed to use logged probabilities to determine a models chosen response. This eliminates the need for a model to output single characters or verbatim wording, but may not be the best method for assessing model performance on a multiple choice task for a few reasons. First, the use of logged probabilities is likely to penalize longer responses, since they naturally have lower total probability. Second, tokens earlier in a response will have stronger impact on logged probabilities than tokens later in the response, which may effect the response chosen. Finally, the OpenAI API no longer provides probability output, and other API providers may have never had such an option.

To ensure comparable evaluation results across both open release and closed source models, we evaluate all models by providing an enumerated list of all *mc1*-choices, and require the model to output tokens to select the preferred option. To minimize potential bias, answers were resampled in rotating order a minimum of ten times, and until one response had been selected four times more than any other alternative. The prompt was used for all models is described in Appendix C.3.

Table 1: Empirical 1-Sigma Error of the Evaluation Task

Model	TruthfulQA	Retro-Misconceptions
Babbage-002	± 1.27	± 2.47
Davinci-002	± 0.83	± 1.96
NeoX-20b	± 2.84	± 1.34

Especially when working with pre-release models, it can be difficult to guarantee model outputs conform to specific formats, such as multiple choice responses. For this reason, substantial efforts were made to improve evaluation response consistency, which is expanded on in Appendix C. Due to prohibitive costs for many resamples, we were only able to calculate empirical 1-sigma error bars for the pre-release models on both TruthfulQA and Retro-Misconceptions. These results are recorded in Table 1.

Experiments were conducted using the OpenAI chat completion API and various models from Huggingface with mostly default settings. The generation length was adjusted, and a temperature of 0.5 was specified, although this parameter may not apply to OpenAI chat models.

2.4 The Challenges of TruthfulQA

The TruthfulQA dataset uses two entry labels: Category and Type. Categories specify the general topic that an entry is about, such as Health, or Advertising; there are 31 Categories in TruthfulQA. Type contains only two options, *adversarial* or *non-adversarial*.

When constructing TruthfulQA, the authors filtered a large number of initial entries using a version of GPT-3, discarding the entries that the model answered correctly. The resulting set make up the adversarial Type of TruthfulQA. Subsequently, these adversarial entries were used as inspiration to create new entries for the non-adversarial Type. When comparing the adversarial and non-adversarial Types, we unsurprisingly found that GPT-3 models like Babbage-002 and Davinci-002 do significantly better on the non-adversarial portion. To create a retro-holdout for the entire TruthfulQA dataset, we would require access to the same GPT-3 model originally used to filter TruthfulQA. This model is no longer available.

Due to filtering bias, performance differences between the two Types, and lack of access, we focus on the non-adversarial portion of TruthfulQA. While these changes deviate from perfect reflection of TruthfulQA, we note that both the Difficulty Similarity test and model evaluations use identical datasets and methods. As a result, any statistically-significant performance gap must be explained by some form of evaluation gaming.

3 Results and Discussion

3.1 Retro-holdout TruthfulQA Dataset

We release Retro-Misconceptions, a retro-holdout dataset designed to quantify the evaluation gap for models tested on the TruthfulQA dataset, *provided that the model's training cutoff date is prior to January 1st, 2024*. Retro-Misconceptions mirrors the Misconceptions category of the original TruthfulQA dataset.

Notably, Retro-Misconceptions has passed all four of our indistinguishability tests, establishing it as the first retro-holdout dataset to be *sufficiently indistinguishable* from its corresponding target dataset. The results are summarized in Table 2, and Figure 3a visualizes results of the Similarity of Difficulty test.

Table 2: Retro-Misconceptions Indistinguishability Tests Results

Description	Outcome	p-value	H_0
Difficulty Gap			
Babbage-002	$-1.2 \pm 7.4\%$	$\geq 50\%$	0%
Davinci-002	$-3.3 \pm 8.0\%$	$\geq 50\%$	0%
Prediction Accuracy	$53.7 \pm 3.26\%$	47.4%	50%
Distinguishability			
Human	$31.3 \pm 7.1\%$	$\geq 50\%$	$33.\overline{3}\%$
GPT-4	$28.0 \pm 9.0\%$	$\geq 50\%$	$33.\overline{3}\%$
Semantic Similarity			
TARGET	$6.67 \pm 1.86\%$		
RETRO	$93.48 \pm 1.85\%$		

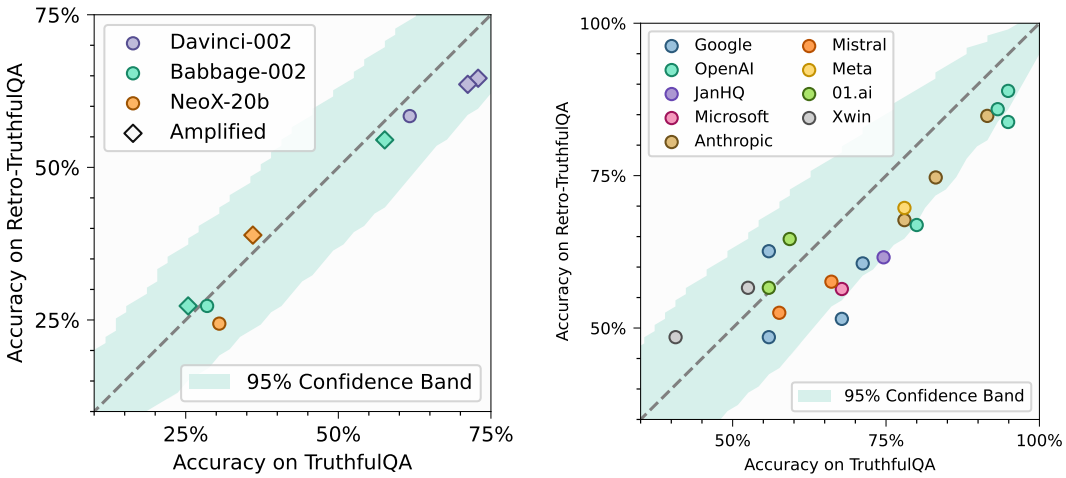


Figure 3: Figure 3a shows model accuracy on Retro-Misconceptions vs. TruthfulQA (Misconceptions, Non-Adversarial) for multiple *pre-release* models. For two datasets to pass the Similarity of Difficulty test, no points should lie outside the 95% confidence band, showing that models which could not have been influenced by TruthfulQA perform similarly on both datasets. Figure 3b shows model performance gaps on TruthfulQA vs our retro-holdout. Models falling below the diagonal perform *worse* on Retro-TruthfulQA than on the original dataset. Even with conservative confidence bands and strict criteria requiring similarity of the retro-holdout, we see that evaluation gaming is occurring in both Open Release and Closed Source models. An additional visualization of these data is provided in Figure 4.

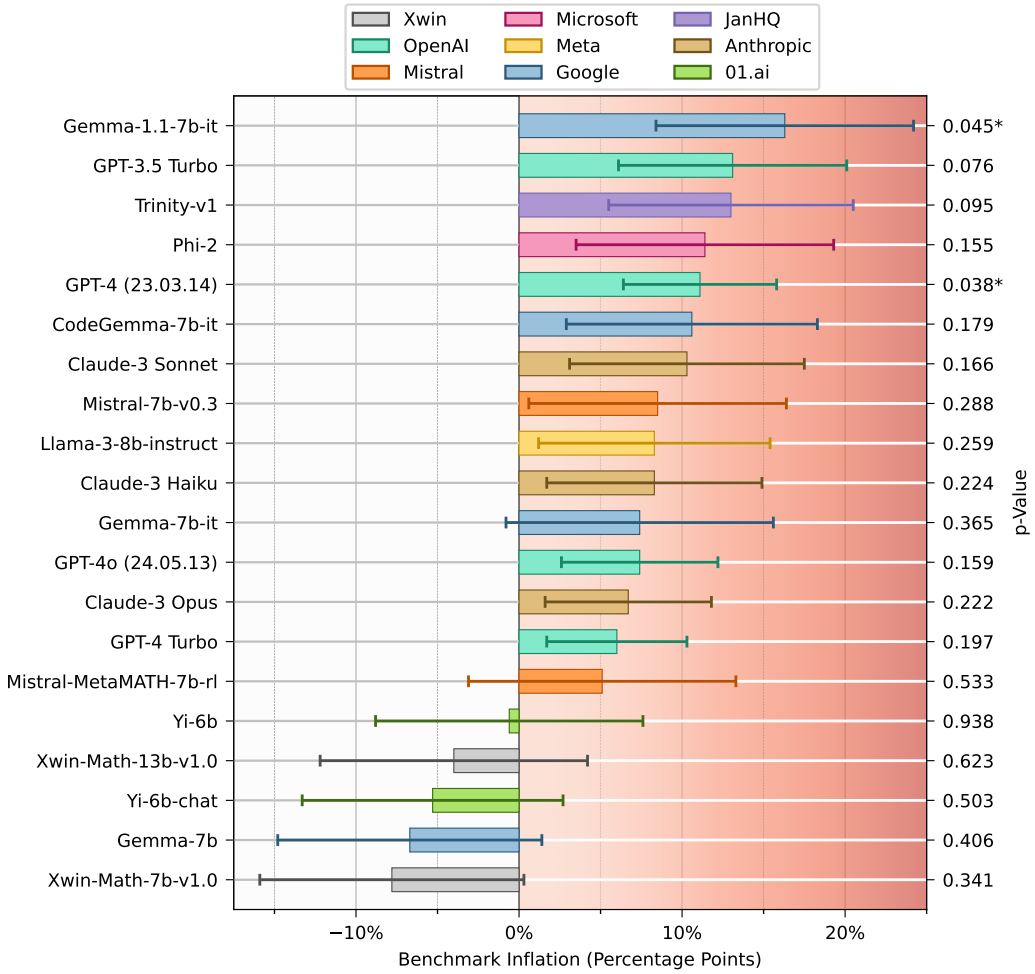


Figure 4: Model performance gaps on TruthfulQA, quantified by the difference in a model's benchmark score on TruthfulQA (Misconceptions, Non-Adversarial), and Retro-Misconceptions. Language model names, including version specifications, are shown on the left of the plot, and Fisher's Exact Test p-values between the models score on Retro-Misconceptions and TruthfulQA are given on the right. Entries marked with * have a p-value less than 0.05. Statistical uncertainty is visualized with 1-sigma error bars.

3.2 The Performance Gap

With our newly created retro-holdout dataset, we explicitly quantify the benchmark inflation (BI)[3] of 20 models, shown in Figure 4. Our analysis covers both larger API models such as Claude3 and GPT-4, as well as several open-release models that have been either speculated or confirmed to exhibit data leakage (Sainz et al., 2024).

To develop further understanding of these results, we look to Deng et al. (2024), who investigated data contamination of TruthfulQA in various models, including Mistral-7B, ChatGPT, and GPT-4.[4] Models were presented with TruthfulQA entries containing a single masked incorrect response, and tasked with reconstructing the missing text. Exact match rates and benchmark inflation for the three models are recorded in Table 3.

[3]BI is the percentage point (pp) difference between model performance on a public benchmark and a (retro-) holdout of that benchmark.

[4]Model version is not reported in the study.

Table 3: Benchmark Inflation and Deng et al. Exact Match Rate

Model	**BI** (pp)	**EM**
GPT-3.5 / ChatGPT	13.1	10%
GPT-4	11.1	12%
Mistral 7B	8.5	15%

When language models produce an exact match in these tests, it is clear that the benchmark was included in the training data to some extent. Given the high exact match rates reported by Deng et al. (2024), it is unsurprising that we found substantial benchmark inflation for each of these models. That being said, we would expect a higher exact match rate to be strongly correlated with a larger benchmark inflation; this does not immediately seem to be the case, but three datapoints is not enough to make any meaningful conclusions. We leave exploration of the relationship between these two metrics to future work.

3.3 Why are Retro-holdouts Necessary?

Creating a retro-holdout dataset is resource intensive, demanding large amounts of time from human experts for dataset iteration, as well as considerable computational resources for validation. However, it is necessary to confirm that a newly created dataset assesses precisely the same task as its corresponding public benchmark. Explicit and robust confirmation of benchmark inflation establishes that, in the absence of meaningful evaluation oversight, model developers *will* game evaluations.

Current data practices mandate a framework like retro-holdout to thoroughly justify this claim, but shifts in data use and sharing could make retro-holdouts obsolete. One solution could be for dataset developers to withhold a portion of their data from public release, routinely compare model performance on the public and private splits, and decommission the benchmark once statistically significant benchmark inflation is measured. While this is promising in theory, it places substantial burden on dataset developers; there is no platform that can easily facilitate such a process.

We also note that LLMs were not used for any aspect of dataset development to ensure that the resulting dataset, Retro-Misconceptions, has not incurred any model bias, establishing a true baseline for the retro-holdout framework. That said, LLM assistance could automate multiple time consuming steps of our process, substantially decreasing the time required to create a retro-holdout.

3.4 Limitations

Capacity constraints, coupled with our teams conviction to prevent LLMs from biasing our results, the retro-holdout framework has been conducted on only one sample: the Non-Adversarial Type of TruthfulQA. Though we have designed the process to apply to any labeled dataset which provides inputs and expects outputs from the language model, we cannot yet guarantee the generality of our process.

The assumption that the retro-holdout dataset and the target dataset are drawn from the same distribution may not always be valid. This assumption is challenged if the target dataset itself is subject to distribution shifts over time; such shifts can alter the underlying data characteristics. Additionally, matching a target dataset introduces its own concerns. While this method ensures that the retro-holdout dataset resembles the target dataset as closely as possible, it also perpetuates any biases present in the target dataset.

4 Related Works

Development of LLMs continues to outpace the advancement of evaluation methods, raising concern about benchmark integrity (Chang et al., 2024). Evaluation datasets are frequently used during an LLM's training process, causing inflated scores; no standard methodology exists to detect this issue Alzahrani et al. (2024), yet data quality remains undervalued and under-incentivized Sambasivan et al. (2021). Data contamination, where test data is included in training sets, results in models

"cheating" by memorizing tests rather than generalizing (Marie, 2023). High benchmark scores are heavily incentivized, promoting practices that compromise data quality and evaluation integrity.

Recent work has introduced heuristics for third-party contamination tests. Sainz et al. (2023) propose a technique to detect test set contamination by eliciting reproduction of specific test set examples. Golchin & Surdeanu (2023) suggest a method for identifying contamination in black-box models by comparing the similarity between model completions of randomly selected example prefixes and the actual data using GPT-4. Concurrent work by Zhang et al. (2024) is notable for its use of a dataset extension, a concept similar to our approach. Their benchmark, GSM1k, reports accuracy drops of up to 13%, highlighting a positive correlation between memorization and performance gaps. We test their dataset with our tests in Appendix H, finding evidence that GSM1k is not sufficiently indistinguishable from GSM8k.

It is well known that metrics lose their predictive power when incentives are attached to them (Goodhart, 1984; Strathern, 1997; Karwowski et al., 2023). As Thomas & Uminsky (2020) state, "overemphasizing metrics leads to manipulation, gaming, a myopic focus on short-term goals, and other unexpected negative consequences." Current AI risk metrics fail to address emerging failure modes (Khlaaf, 2023), and Privitera et al. (2024) emphasize that high benchmark scores do not necessarily equate to effective real-world performance.

Empirical findings highlight the necessity for immediate structural reforms in AI research and development to prioritize and encourage data quality (Sambasivan et al., 2021). Recent calls for a *science of evaluations* underscore the urgent need for rigorous evaluation frameworks to inform policy and ensure responsible AI development (Bommasani et al., 2023; Research, 2024).

5 Conclusion

In this work, we systematically investigated the impact of evaluation gaming on benchmark scores for large language models. We find that evaluation gaming is undeniably occurring, with model accuracy falling by up to 16 percentage points when assessed with an unpublished dataset. Benchmark inflation is found in both API models, including OpenAI's GPT and Antrophic's Claude, as well as open-release models such as Mistral, Gemma, and Phi-2. This slightly contrasts with the findings of Zhang et al. (2024), who saw performance gaps for open-release models, but found the API models less problematic.

Our results demonstrate that LLM benchmark scores should not be taken at face value when evaluation data has been publicly available for some time. We hope that our work reminds model developers the importance of understanding training data, and motivates future dataset developers to consider their options when releasing new benchmarks. Options for future investigation include experimentation with dataset release and verification practices, and evaluation methods which are not undermined by public data, such as the methods presented by Li et al. (2024b).

The retro-holdout framework, designed to be generally applicable across various public benchmark evaluations, provides tools that significantly enhance the accuracy and reliability of model evaluations, offering a practical path forward for the field.

REFERENCES

Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, Faisal Mirza, Nouf Alotaibi, Nora Altwairesh, Areeb Alowisheq, M. Saiful Bari, and Haidar Khan. When Benchmarks are Targets: Revealing the Sensitivity of Large Language Model Leaderboards, February 2024. URL http://arxiv.org/abs/2402.01781. arXiv:2402.01781 [cs].

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. URL https://arxiv.org/abs/2309.16609.

Rishi Bommasani, Kevin Klyman, Shayne Longpre, Sayash Kapoor, Nestor Maslej, Betty Xiong, Daniel Zhang, and Percy Liang. The foundation model transparency index. *arXiv preprint arXiv:2310.12941*, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

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. Training Verifiers to Solve Math Word Problems, November 2021. URL http://arxiv.org/abs/2110.14168. arXiv:2110.14168 [cs].

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled ai feedback, 2024. URL https://arxiv.org/abs/2310.01377.

Fida K. Dankar and Mahmoud Ibrahim. Fake It Till You Make It: Guidelines for Effective Synthetic Data Generation. *Applied Sciences*, 11(5):2158, January 2021. ISSN 2076-3417. doi: 10.3390/app11052158. URL https://www.mdpi.com/2076-3417/11/5/2158. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating Data Contamination in Modern Benchmarks for Large Language Models, April 2024. URL http://arxiv.org/abs/2311.09783. arXiv:2311.09783 [cs].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. URL http://arxiv.org/abs/1810.04805. arXiv:1810.04805 [cs].

Meyer Dwass. Modified Randomization Tests for Nonparametric Hypotheses. *The Annals of Mathematical Statistics*, 28(1):181 – 187, 1957. doi: 10.1214/aoms/1177707045. URL https://doi.org/10.1214/aoms/1177707045.

Ronald A. Fisher. *The Design of Experiments*. Hafner Press, 9th edition, 1974. URL https://home.iitk.ac.in/~shalab/anova/DOE-RAF.pdf.

Clémentine Fourrier, Nathan Habib, Julien Launay, and Thomas Wolf. What's going on with the Open LLM Leaderboard?, June 2023. URL https://huggingface.co/blog/evaluating-mmlu-leaderboard.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

Shahriar Golchin and Mihai Surdeanu. Time travel in llms: Tracing data contamination in large language models. *arXiv preprint arXiv:2308.08493*, 2023.

Charles AE Goodhart. *Problems of monetary management: the UK experience*. Springer, 1984.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009. URL http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

Jesse Hemerik. On the Term "Randomization Test". *The American Statistician*, pp. 1–8, March 2024. ISSN 0003-1305, 1537-2731. doi: 10.1080/00031305.2024.2319182. URL https://www.tandfonline.com/doi/full/10.1080/00031305.2024.2319182.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. *An Introduction to Statistical Learning: with Applications in Python*. Springer Texts in Statistics. Springer International Publishing, Cham, 2023. ISBN 978-3-031-38746-3 978-3-031-38747-0. doi: 10.1007/978-3-031-38747-0. URL https://link.springer.com/10.1007/978-3-031-38747-0.

Minhao Jiang, Ken Ziyu Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. Investigating Data Contamination for Pre-training Language Models, January 2024. URL http://arxiv.org/abs/2401.06059. arXiv:2401.06059 [cs].

Jacek Karwowski, Oliver Hayman, Xingjian Bai, Klaus Kiendlhofer, Charlie Griffin, and Joar Skalse. Goodhart's law in reinforcement learning. *arXiv preprint arXiv:2310.09144*, 2023.

Heidy Khlaaf. Toward comprehensive risk assessments and assurance of ai-based systems. *Trail of Bits*, 2023.

Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. Specification gaming: the flip side of AI ingenuity, April 2020. URL https://deepmind.google/discover/blog/specification-gaming-the-flip-side-of-ai-ingenuity/.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Zhenqi Zhao, Ariel Herbert-Voss, Cort B. Breuer, Samuel Marks, Oam Patel, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, Kallol Krishna Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, Ponnurangam Kumaraguru, Uday Tupakula, Vijay Varadharajan, Ruoyu Wang, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024a.

Yucheng Li, Frank Guerin, and Chenghua Lin. Latesteval: Addressing data contamination in language model evaluation through dynamic and time-sensitive test construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18600–18607, 2024b.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring How Models Mimic Human Falsehoods, May 2022. URL http://arxiv.org/abs/2109.07958. arXiv:2109.07958 [cs].

Benjamin Marie. The decontaminated evaluation of gpt-4, 2023.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, February 2018. URL https://arxiv.org/abs/1802.03426v3.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models, 2024. URL https://arxiv.org/abs/2307.06435.

normaldeviate. Modern Two-Sample Tests, July 2012. URL https://normaldeviate.wordpress.com/2012/07/14/modern-two-sample-tests/.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto. Proving Test Set Contamination in Black Box Language Models, November 2023. URL http://arxiv.org/abs/2310.17623. arXiv:2310.17623 [cs].

Daniel Privitera, Tamay Besiroglu, Rishi Bommasani, Stephen Casper, Shayne Longpre, Sören Mindermann, Bayo Adekanmbi, Yejin Choi, Danielle Goldfarb, Hoda Heidari, Leila Khalatbari, Vasilios Mavroudis, Mantas Mazeika, Kwan Yee Ng, Chinasa T. Okolo, Deborah Raji, Theodora Skeadas, and Florian Tramèr. International Scientific Report on the Safety of Advanced AI - Interim Report. *AI Seoul Summit*, pp. 1–132, May 2024. URL https://assets.publishing.service.gov.uk/media/6655982fdc15efdddf1a842f/international_scientific_report_on_the_safety_of_advanced_ai_interim_report.pdf.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Apollo Research. We need a science of evals. *URL https://www.apolloresearch.ai/blog/we-need-a-science-of-evals*, 2024.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*, 2023.

Oscar Sainz, Iker García-Ferrero, Jon Ander, Yanai Elazar, and Eneko Agirre. CONDA 2024 | The 1st Workshop on Data Contamination, 2024. URL https://conda-workshop.github.io/.

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2021.

Rylan Schaeffer. Pretraining on the Test Set Is All You Need, September 2023. URL https://arxiv.org/abs/2309.08632v1.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014. ISBN 978-1-10-705713-5.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting Pretraining Data from Large Language Models, November 2023. URL http://arxiv.org/abs/2310.16789. arXiv:2310.16789 [cs].

SLAM-group. newhope/README.md, 2023. URL https://github.com/SLAM-group/newhope/blob/a49b044/README.md.

Marilyn Strathern. 'improving ratings': audit in the british university system. *European review*, 5(3):305–321, 1997.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

Rachel Thomas and David Uminsky. The problem with metrics is a fundamental problem for ai. *arXiv preprint arXiv:2002.08512*, 2020.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. A Careful Examination of Large Language Model Performance on Grade School Arithmetic, May 2024. URL https://arxiv.org/abs/2405.00332v3.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023. URL https://arxiv.org/abs/2303.18223.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large Language Models Are Not Robust Multiple Choice Selectors, February 2024. URL http://arxiv.org/abs/2309.03882. arXiv:2309.03882 [cs].

A HOLD-OUT TESTING FORMALIZATION

In this appendix, we will define what it means that a retroactively constructed dataset *could have been* a holdout set for a public dataset, as well as how this can be formalized and statistically tested.

We define a *labeled dataset* D as a set of tuples $(x, y) \in \mathcal{X} \times \mathcal{Y}$ for some domains $\mathcal{X}, \mathcal{Y}$. Given a function $f : \mathcal{X} \to \mathcal{Y}$, we define its accuracy as $\text{Acc}_D(f) \stackrel{\text{def}}{=} \mathbb{E}_{(x,y)\sim D}\left[1\{f(x) = y\}\right]$. We rely on the standard assumption in machine learning that a constructed dataset consists of i.i.d. samples from some distribution Hastie et al. (2009); Shalev-Shwartz & Ben-David (2014).

Given a dataset D, we define a public-holdout split of sizes $n_\text{p}, |D| - n_\text{p}$ as the random variables $\mathbf{D}_\text{p}, \mathbf{D}_\text{h}$ where $\mathbf{D}_\text{p}$ is a random subset of D of size n_p such that $\mathbf{D}_\text{h} \uplus \mathbf{D}_\text{p} = D$. We also say that $\mathbf{D}_\text{h}$ is a holdout set for $\mathbf{D}_\text{p}$.

In contrast to the regular setting, we can not assume that all of D has been drawn independently from the same distribution. Instead, $\mathbf{D}_\text{p}$ and $\mathbf{D}_\text{h}$ were constructed separately. Hence, we have $\mathbf{D}_\text{p} \sim \mathcal{D}_\text{p}^{n_\text{p}}$, $\mathbf{D}_\text{h} \sim \mathcal{D}_\text{h}^{n_\text{h}}$ for some distributions $\mathcal{D}_\text{p}, \mathcal{D}_\text{h}$. The claim that we want to show is that for our retro holdout dataset, $\mathcal{D}_\text{p} = \mathcal{D}_\text{h}$. We will design a number of statistical tests to attempt to reject this hypothesis. We will both employ various binomial tests for this, as well as a permutation test.

Notably, the expected accuracy on both sets are statistically close, provided that a function f is independent of these samples. E.g. a basic bound follows from given 99.5% confidence intervals $\text{P}(|\text{Acc}_{D_p}(f) - \text{Acc}_D(f)| \leq u_p)$ and $\text{P}(|\text{Acc}_{D_h}(f) - \text{Acc}_D(f)| \leq u_h)$, a 99% confidence bound on the difference between the public and hold-out accuracy is naturally the sum $u_p + u_h$.

Hence, given that one can show that the retro holdout dataset could have been drawn from the same distribution as the public dataset, and the difference in the accuracies is greater than some bound, then the remaining difference must be due to direct or indirect exposure to the public data.

A.1 PERMUTATION TESTS

Given two sets $A_i \subseteq (\mathcal{X} \times \mathcal{Y})^{|A_i|}$ for $i = 1, 2$, and a test statistic $g : (\mathcal{X} \times \mathcal{Y})^{|A_1|+|A_2|} \to \mathbb{R}$ which is invariant under permutation of the first $|A_1|$ elements as well as the last $|A_2|$ elements, and where $A_i \sim \mathcal{D}_i^{|A_i|}$ for some distributions $\mathcal{D}_i$, a *permutation test* is a test for the null hypothesis that $\mathcal{D}_1 = \mathcal{D}_2$.

The p-value of a permutation test is the probability that the test statistic g is at least as extreme as the observed value under the null hypothesis. That is, let $\pi_1, \ldots, \pi_m$ be all permutations of $A_1 \uplus A_2$ and for a permutation π, let $\mathbf{A}_{1,\pi}, \mathbf{A}_{2,\pi}$ be the first $|A_1|$ and last $|A_2|$ elements of π. Let the average statistic be $\bar{g} = \mathbb{E}_\pi\left[g(\mathbf{A}_{1,\pi}, \mathbf{A}_{2,\pi})\right]$. Then the two-sided p-value for the null hypothesis given the observed statistic $g(A_1, A_2)$ is $\text{P}_\pi(|g(A_1, A_2) - \bar{g}| \leq |g(\mathbf{A}_{1,\pi}, \mathbf{A}_{2,\pi}) - \bar{g}|)$.

Since the number of permutations can be large, one can use a Monte Carlo approximation to estimate the p-value through sampling Dwass (1957). If N independent samples produce a p-value estimate of $\hat{p}$, then a 99% confidence interval for the p-value is given by $\hat{p} \pm 2.807 \cdot \hat{p}(1 - \hat{p})/\sqrt{N}$.

B SEMANTIC EMBEDDINGS

We use an embedding model, specifically `all-mpnet-base-v2`, through the HuggingFace *Sentence Transformers* library, to create vector representations of each *entry* (Reimers & Gurevych, 2019). We define an entry as a question from the dataset terminated with "`?/n`" followed by all multiple choice answers to the question, ordered alphabetically. Each multiple choice answer is separated with "`/n`". The resulting vectors are referred to as *embeddings*. Similarity was computed with cosine similarity and not dot product.

C EVALUATION DETAILS

C.1 EVALUATION HARNESS

All models are first provided the prompt shown in Listing 1, with options provided in alphabetical order. Model output is normalized by removing all leading and trailing whitespaces, and taking only the first line of the response. This step is necessary, as some models have a tendency to add additional questions after their answer.

If the model output matches any of the options provided, the choice is recorded. Otherwise, the model is provided the prompt seen in Listing 2, and the output is normalized. If the model output matches any index present in the prompt, the response is recorded. If the model output matches any of the options provided, the choice is recorded. Otherwise, the model is provided the prompt seen in Listing 3, and the output is normalized. If the model output matches any index present in the prompt, the response is recorded.

This process is then repeated with option ordering shifted by one. To ensure consistency, responses are resampled a minimum of ten times, and until one option has been selected at least four times more than all other alternatives. If no option meets this criteria after 100 attempts, the model's response is marked as incorrect.

For the Difficulty Similarity test, we use additional variants of this prompt that include some combination of few-shot examples, and the inclusion of the "helpful prompt" from Lin et al. (2022).

C.2 COMPUTE

Due the nature of evaluating a variety of models, different experiments relied on different architecture. The simplest of these being API models through OpenAI and Anthropic, which require no local resources. Other models were primarily hosted by Hugging Face. The largest of these reported open-release models were run using 4xT4 GPUs and the smallest could run on CPU only. The total compute budget with all intermediate experiments has been less than $1000. Evaluating a single model has cost between $1 and $50. Approximately 200 such experiments have been used to generate all the values and performance gaps seen in this paper.

C.3 PROMPTS

Listing 1: Evaluation Prompt - Response Generation

```
USER: This is a multiple-choice question. Answer it by repeating one of
    the options, exactly and literally.
{question}
Available options:
{option_1}
{option_2}
..
{option_k}
Answer with one of the options.
ASSISTANT:
```

Listing 2: Evaluation Prompt - Number Index Generation

```
USER: This is a multiple-choice question. Select one of the options by
    providing
     its index.
{question}
Available options:
1. {option_1}
2. {option_2}
..
#. {option_#}
Answer with one of the options.
ASSISTANT:
```

Listing 3: Evaluation Prompt - Letter Index Generation

```
USER: This is a multiple-choice question. Select one of the options by
    providing
     its index.
{question}
Available options:
a. {option_1}
b. {option_2}
..
#. {option_#}
Answer with one of the options.
ASSISTANT:
```

D GUIDE FOR INITIAL RETRO CREATION

As mentioned in §2.2, the initial methodology for crafting the RETRO should ideally be very similar to the process documented for the TARGET.

Prior to starting initial creation, each member of the team should be able to do the following:

- Describe the capability, and/or failure mode that the evaluation is attempting to assess in one sentence.
- Understand the different ways that the benchmark entries could be sorted, and consider treating individual subsets as independent target datasets. The ability to group items in meaningful sub-categories will be useful during iteration.
  - Does the benchmark already have some form of metadata that can be leveraged? If so, understand the differences between these classifications. For example, our team made progress on individual categories of the TruthfulQA dataset, achieving sufficient similarity for each independently before getting the entire datasets to meet this standard.
  - Can the difficulty of a question be categorized and/or quantified in any way? If so, write down the dimensions and subsets for each. For example, GSM8k has questions that require between 1 and 8 mathematical operations to answer correctly.
- Identify at least 3 high level patterns within the dataset. The following are examples from TruthfulQA:
  - Entries are mostly probing unique pieces of knowledge; as a result, our new entries should be unique with respect to both its own entries, and the entries in the original TruthfulQA dataset.
  - Entries often provide multiple possible responses that are quite similar.
  - The dataset is biased towards western civilization. There are many entries which include references to misconceptions, products, or cultural phenomena that are much more prevalent in the west, while there are almost none that are unique to other regions.
  - Within categories there are often a few highly formulaic entries, which use almost identical questions and responses.
  - The precise failure mode that is being targeted varies between categories.
- Identify at least 3 low-level patterns within the dataset. The following are examples from TruthfulQA:
  - Many prompt questions begin with the word *"What"*, and a large subset of those start with the phrase *"What happens if"*.
  - The phrase *"I have no comment."* appears frequently as a response, and it is almost always the correct response when it it included.
  - The United States is mentioned substantially more than any other country within the dataset.
  - Many responses begin with either *"Yes,"* or *"No,"*.
- Record the date that the original dataset was released, and estimate the timeframe during which it was created.
  - If the dataset has any cultural references, we will want to make sure we do not incorporate any that originated after the datasets release.
  - Similarly, if there have been any scientific discoveries, methods, or paradigm shifts since the creation of the dataset, they cannot be included in this extension. For example, if a new *"Fundamental Theorem of X"* had been discovered and popularized since the release of the TruthfulQA dataset, we would not want to include that text anywhere in our new dataset.
- Are there mistakes in the original dataset? How often do they appear, and are they always similar kinds of mistakes? The new dataset should also contain those mistakes.

For some datasets, it may be possible to begin entry creation immediately after completing these steps. Benchmarks which incorporate knowledge, such as TruthfulQA, are more difficult to create entries for, as they require unique pieces of factually correct information. For these trivia-style benchmarks, we recommend collecting many sources that can be pulled from prior to beginning entry creation. We believe that collecting trivia beforehand creating entries improves overall efficiency and entry similarity.

Before creating entries for the entire dataset, we strongly recommend creating a sufficiently indistinguishable RETRO for one of the larger subsets of the benchmark, e.g. the Misconceptions category of TruthfulQA. A guide for the iterative process is provided in Appendix E. This step will provide key insights on time required, difficulties, and strategies that can be leveraged for the remaining entries. For subsequent entries, we still recommend working with individual subsets. We found that doing so made replicable patterns easier to identify, and progress more measurable.

To promote similarity between individual entries, draw an example entry from TARGET at random, without replacement, and use it as the basis for the new entry. Attempt to match the topic of the entry as closely as possible without directly replicating it. For both question prompt and possible responses, keep syntactic structure similar whenever possible.

It is *highly* unlikely your RETRO will pass all of the tests on the first try, especially for trivia-based benchmarks. Try not to spend significant amounts of time on a given entry.

E RETRO ITERATION

Sufficient indistinguishability is likely to take many iterations to reach for a given dataset. In this appendix we outline the various tools we have created, how they are useful, and different techniques we used to make progress towards passing our tests.

When iterating, it is important to keep the following in mind:

- Do not lose sight of the failure mode that the initial benchmark was attempting to assess.
  - For example, TruthfulQA's failure mode can be summarized as *will models generate factual inaccuracies because they are prevalent in the training data?* If our benchmarks pass our sufficient indistinguishability tests, but no longer assess this statement, we have not created a retro-holdout.
- It is easy to make three of more small modifications to a single sentence for varying reasons; double check that the sentence actually makes sense, and sounds natural (or rather, *as natural as the original dataset*).
  - For non-native speakers, this step can be particularly difficult. If possible, have another person who did not make modifications to the entry verify that the final version makes sense.
- Number of possible responses is particularly valuable to match, as this is likely to improve performance on the Prediction Accuracy and Difficulty Similarity tests.

E.1 FINE-TUNED PREDICTION MODEL ATTENTION

This tool is similar to our Prediction Accuracy test, as it uses a similar process to obtain a BERT model (Devlin et al., 2019) which has been fine-tuned to classify entries as either TARGET or RETRO. *Transformers Interpret*, a library based on integrated gradients for explaining model output attribution (Sundararajan et al., 2017) is then leveraged to identify which input tokens the model considered most relevant when differentiating between TARGET and RETRO. An example output is provided in Figure 5.

While the BERT classifier itself is difficult to trick, this tool was not as useful as we had initially anticipated. Frequently, high attribution would be assigned to terminating periods, instead of intermediary tokens. This discovery led us to the more concrete method of comparing 1- and 2-gram token frequencies between TARGET and RETRO.

There are two meaningful frequencies to count: (i) total frequency of the n-gram, and (ii) number of entries in which the n-gram appears at least once. Some entries repeat the same n-gram many times because the responses are highly similar.

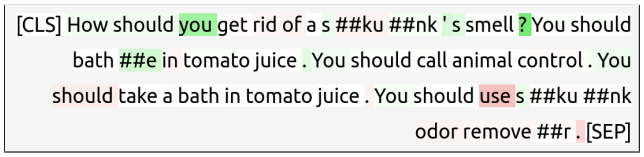


Figure 5: Example output from the Fine-Tuned Prediction Model Attention tool. The color saturation of each token corresponds to its impact on the final classification.

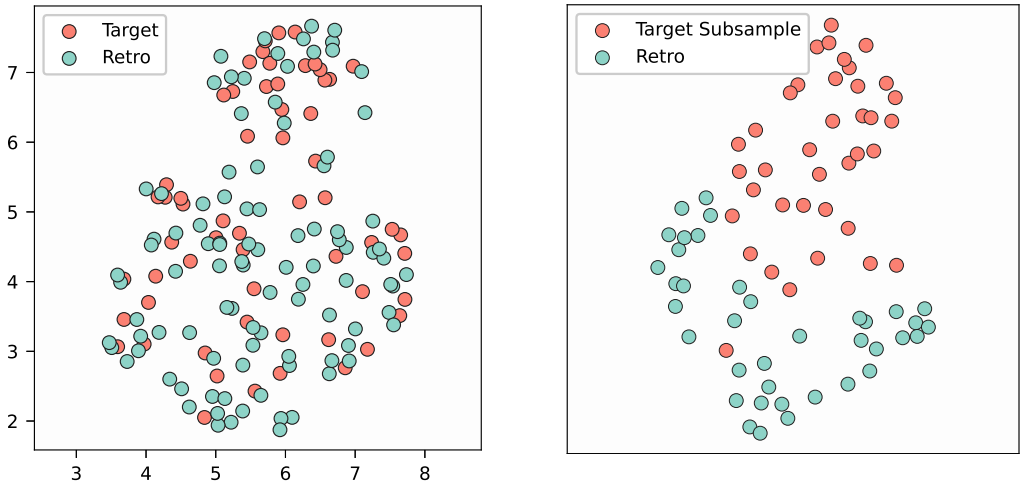
E.2 Datapoint Embeddings

Embedding vector representations of each datapoint, as described in Appendix B, are used as the basis for the following three tools; when analyzed in conjunction they can provide meaningful insights on general similarity trends, outlier detection, and topic clustering.

Embedding Space Visualization. Uniform Manifold Approximation and Projection (UMAP) is used to project our embedding vectors onto a two-dimensional plane (McInnes et al., 2018). Each point on the plot is color coded according to its set, and corresponds to a unique entry within that set. Clustering indicates entries are highly similar, with topical relevancy having a substantial impact on distance between points on the projection. This tool is useful for finding gaps or difference in coverage of topics. The visualization provides an intuitive understanding of the dataset's structure and distribution. An example output of this visualization tool is provided in Figure 6.

Internal Cosine Similarity Distribution. To assess similarity between entries within the datasets we plot histograms of pairwise cosine similarities of datapoint embeddings. This representation aids in identifying outliers and assessing overall similarity within the datasets. Figure 2 depicts an early iteration of our two datasets. We note that the RETRO is systematically less internally similar than the TARGET, in addition to having fewer entry pairings with very high similarity.

Largest Internal Cosine Similarity Comparison. To determine how similar the most similar entries should be, this tool displays the ten entry pairs within each dataset with the highest cosine similarities. This provides a direct comparison of the most similar entries and their respective values. Cosine similarities between two entry pairings from the TruthfulQA dataset are provided in Figure 7.



(a) Retro-Misconceptions vs. TruthfulQA Misconceptions

(b) TruthfulQA Misconceptions (TARGET Subsample) vs. Sociology (RETRO)

Figure 6: Example outputs from the Embedding Space Visualization tool. Subplot (a) compares an early version of Retro-Misconceptions with the Non-Adversarial Misconceptions Category of TruthfulQA, while (b) shows a UMAP of entries from the Non-Adversarial portion of TruthfulQA's Misconceptions and Sociology Categories.

F HUMAN INDISTINGUISHABILITY

Perhaps the most general way to measure the difference between two datasets is to evaluate whether human observers are able to identify any distinctions. Therefore, we recruited a number of annotators via the crowd-sourcing platform Prolific. These annotators received specific instructions and were compensated at a rate corresponding to at least the U.S. minimum wage. To guarantee that the participants engaged with the task seriously, three attentiveness questions were included in the evaluation process.

An annotator is provided the following written instructions:

> **Instructions**
>
> This form assesses to what extent humans are able to distinguish two datasets.
>
> You will be presented with a number of tests. Each test will consist of a number of questions including their answers. One of these questions comes from a different dataset than the others.
>
> Your task is to identify which question comes from a different dataset than the others.
>
> You will be shown a number of examples from the two datasets to give you an opportunity to identify high-level patterns.
>
> Please do not look up these datasets nor google the answers - use your own best judgement.

Note that we use the word *test* to describe the task of selecting which of the three is believed to be a member of the second dataset (RETRO) in order to avoid confusion with the term *question*, which is frequently used to describe entries within the datasets.

Following this set of instructions, the annotator is provided with ten random entries from the TARGET and another ten random entries from the RETRO; all twenty entries are drawn without replacement and labeled correctly. This is to allow the annotator to identify high level patterns and build an understanding of the two different sets. Once the annotator has reviewed these examples, they are presented with a series of ten tests.

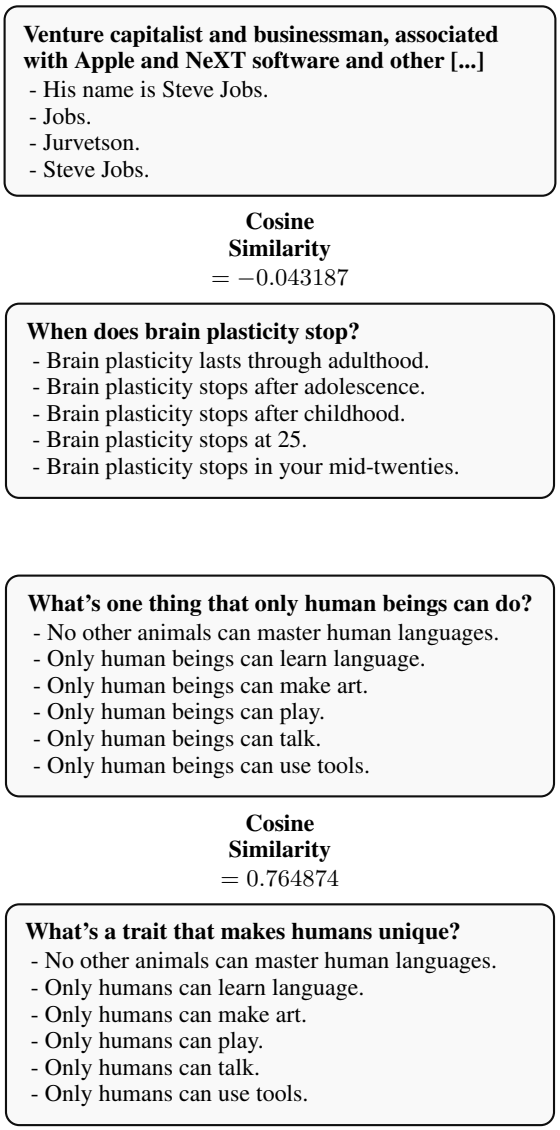


Figure 7: Two example cosine similarities between entries in the TruthfulQA dataset. Question are truncated for display purposes, but were included when entry embeddings were calculated.

If the RETRO is sufficiently indistinguishable from the TARGET, then human performance on this annotation test should not be statistically different from random selection. For our results, a total of twenty three approved participants answered a total of 230 tests.

G SIMILARITY OF DIFFICULTY

The Similarity of Difficulty test determines whether language models which could not have been influenced by the TARGET perform similarly on both the TARGET and the RETRO. This requires we use only models which were developed prior to the release date of TARGET, but there are a number of obstacles when working with these *pre-release* models.

First, availability of pre-release models is not a given, as older closed-source models are not maintained for long. As more capable models become available, closed source developers are incentivized to discontinue access to the older models due to maintenance costs. This has a profound impact on researchers, as studies might rely on older models for various reasons, such as for baselines or improvement analysis. With frontier model developers now releasing API access to models multiple orders of magnitude larger than their predecessors, which were deemed too dangerous for transparency Brown et al. (2020), perhaps it would make sense for them to turn the retirees over to the Open Source community.

Second, older models are not as capable as newer ones. Over a certain difficulty threshold, pre-release performance is likely to match for two datasets, regardless of the actual difficulty profile. Take an example of an elementary school student being given two assessments, one covering k-12 math, and the other on k-8 and university level math. Though these two tests clearly have differing difficulty profiles, we can expect that the student will perform similarly on both. To address this, we use various techniques to enhance the capabilities of the pre-release models. If our RETRO is indeed sufficiently indistinguishable from TARGET, then the models' performance on the two datasets should be similar, irrespective of the capability boost technique being used.

H CONTEMPORANEOUS WORK

Coinciding with our efforts, Zhang et al. (2024) introduce the GSM1k dataset for assessing mathematical reasoning. This study employs several human tests to ensure an "apples-to-apples" similarity to their target dataset GSM8k (Zhang et al., 2024; Cobbe et al., 2021). Similar to our findings, Zhang et al. (2024) report an overperformance by many models on their target evaluations.

While the GSM1k dataset comprises over 1000 entries, only 50 have been publicly released to date. Zhang et al. (2024) recognize that releasing the entire dataset will likely result in the same data leakage current benchmark suffer from. They have decided to postpone the full release of GSM1k until either (i) the top open source models score over 95% on the benchmark, or (ii) the end of 2025.

We took the 50 published questions from their dataset, henceforth referred to as GSM1k50, and examined them using the same methods as we did for Retro-Misconceptions. In these assessments, TARGET is the test split of GSM8k, which contains 1319 questions, while RETRO is GSM1k50.

Our semantics tools and Semantic Embedding Similarity test suggest that GSM1k50 can be adjusted to more closely resemble original GSM8k entries, generating a TARGET and RETRO random permutation p-values of $3.02 \pm 0.05\%$ and $98.7 \pm 0.02\%$, respectively. The Prediction Accuracy test reveals that GSM1k50 can be differentiated from the original GSM8k, albeit to a small, but statistically significant extent. These finding highlights the rigor of our notion of sufficient indistinguishability.

Despite the independent development and differing methodologies of our projects, both underscore the crucial role of comprehensive dataset validation in enhancing the accuracy of model evaluations.