Benchmark Inflation: Revealing LLM Performance Gaps Using Retro-Holdouts

Anonymous Author(s)
Affiliation
Address
email

Abstract

Public benchmarks are compromised, as the training data for many Large Language Models (LLMs) is contaminated with test data, 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-TruthfulQA, 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.

1 Introduction

Concerns have emerged about the reliability of public benchmarks to accurately assess the performance of large language models [1, 56, 15]. First, there is a notable discrepancy between the reported performance of models on evaluation datasets and their actual capabilities in practical settings [33]. Second, achieving high scores on these evaluations is strongly incentivized, as higher scores are closely linked to increased publicity and wider adoption of the given model [24]. This emphasis on benchmarks fosters a competitive environment where optimizing for benchmark performance can take precedence over real-world performance, potentially compromising the practical effectiveness or safety of models. This situation resembles specification gaming, where models meet the requirement of scoring well on benchmarks without genuinely improving on the capabilities that these benchmarks aim to assess [29]. Extending this framing, we define the mechanisms leading to a systematic gap between benchmark scores and real-world performance as *evaluation gaming*.

Recent research has shown that evaluation datasets have, in some cases, been included in the training data [43, 38, 47, 49, 26, 50], demonstrating that evaluation gaming is occurring. Such data leakage can destroy the predictive power of benchmarks, leading to large performance gaps between a model's evaluation scores and its actual performance, as well as undermining trust in the reported model scores [39] – this highlights the need to improve practices for dataset release, and data collection. Such issues are particularly problematic given the significant role that evaluations are likely to play in the governance of machine learning technologies; stronger economic incentives will only increase the likelihood and severity of evaluation gaming. Furthermore, by misrepresenting model capabilities, current evaluations may create a false sense of safety. To accurately gauge the difference in a model's

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

performance between the specific evaluation task and an analogous real-world task, we need access to a dataset originating from the same data distribution that has not been used during model training.

This is the idea of *holdout* datasets, which are used to assess a machine learning model's 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 [25]. Systematic differences in performance between holdout and target datasets can point to overfitting caused by data leakage. Comparing a model's performance on a public benchmark and a corresponding holdout dataset could reveal whether data from the public benchmark has influenced the training process. Unfortunately, holdout datasets are typically not available; benchmark developers usually release all evaluation data, although there are notable exceptions, e.g. Li et al. [31].

To address these challenges, we propose *retroactive holdout*, or *retro-holdout*, datasets, which are verified to be 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. Our research advances the field by introducing a general and scalable methodology to create a retro-holdout dataset for a fully disclosed evaluation dataset, followed by rigorous testing to verify that the retro-holdout dataset closely mirrors the target dataset.

We detail our methodology for generating and validating retro-holdout datasets, along with recommendations and tools. We conduct a demonstrative case study using the TruthfulQA benchmark [34], a question answering dataset that was designed to assess the propensity of language models to mimic human falsehoods. TruthfulQA was selected for two key reasons: (i) it has become a popular dataset for developers to test against [32] and (ii) it has clear safety implications, as models performing poorly are likely to respond to user input with believable falsehoods.

1.1 Contributions

In this work, we:

- Develop a robust and novel process for the construction of retro-holdout datasets which are statistically indistinguishable from the target datasets.
- Introduce four tests for determining the similarity between two evaluation datasets, enabling identification of appropriate retro-holdout datasets for accurate model evaluations.
- Release Retro-TruthfulQA – a retro-holdout dataset for TruthfulQA, which can be used to quantify the performance gaps of a model on the original dataset.[1]
- Conduct a comprehensive evaluation of 20 models using Retro-TruthfulQA to demonstrate measurable score inflation.

2 Methods

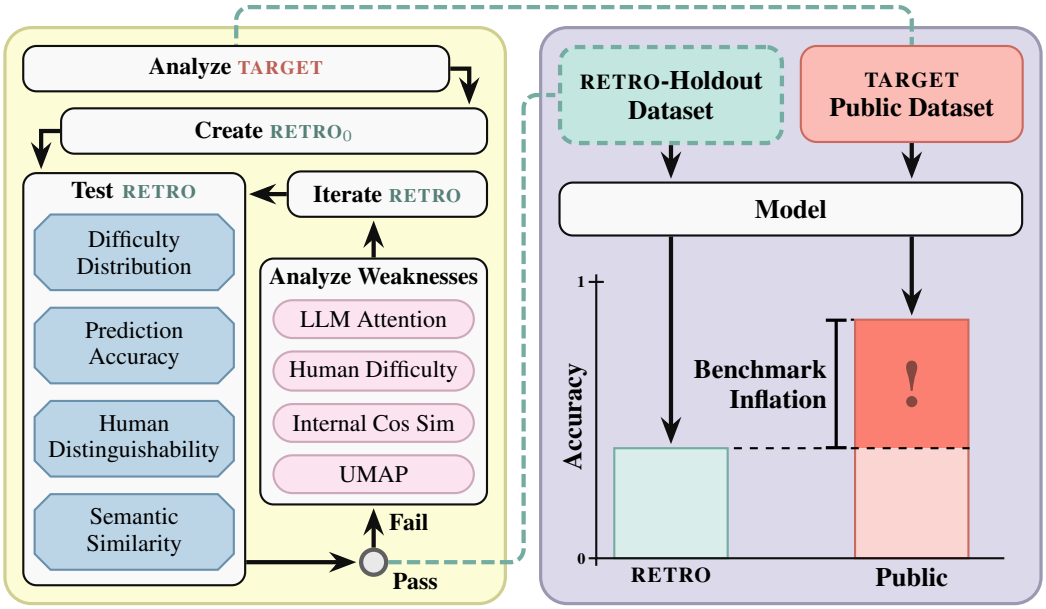
Holdout datasets were first used in machine learning to accurately assess model performance. Unlike conventional holdout sets, retro-holdout datasets are not just randomly selected subsets; they are independently created post-hoc to match the statistical properties of the target dataset, thereby ensuring that they serve as effective and unbiased benchmarks for assessing real-world performance of the model post-training.

For brevity, we define

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

We assume that the entries in TARGET were drawn a parent distribution which we denote as PARENT. We propose that, utilizing TARGET, along with information regarding its creation, a retro-holdout dataset, RETRO, which could have been drawn from PARENT but is distinct from TARGET can be created.

[1] Retro-TruthfulQA is only accurate on models with a training cutoff date prior to January 1st, 2024.



Create and validate Retro-Holdout

Quantify performance gap using Retro-Holdout

Figure 1: Visualization of our methodology.

2.1 Creating the RETRO

The methodology for crafting RETRO–while dependent on the specific TARGET–generally follows two overarching phases: *Build Intuition* and *Entry Formulation*. Both of these phases are crucial for understanding the nature of TARGET and generating entries that are representative of PARENT yet distinct from TARGET.

Build Intuition To create a robust RETRO, one must have a strong understanding of the TARGET, focusing primarily on its intended purpose and the methodology of its creation. We recommend an initial thorough review of the dataset documentation and relevant literature, as well as looking at many entries within TARGET. This phase, though straightforward, has proven to yield critically valuable insights for the subsequent formulation, and later iteration, processes.

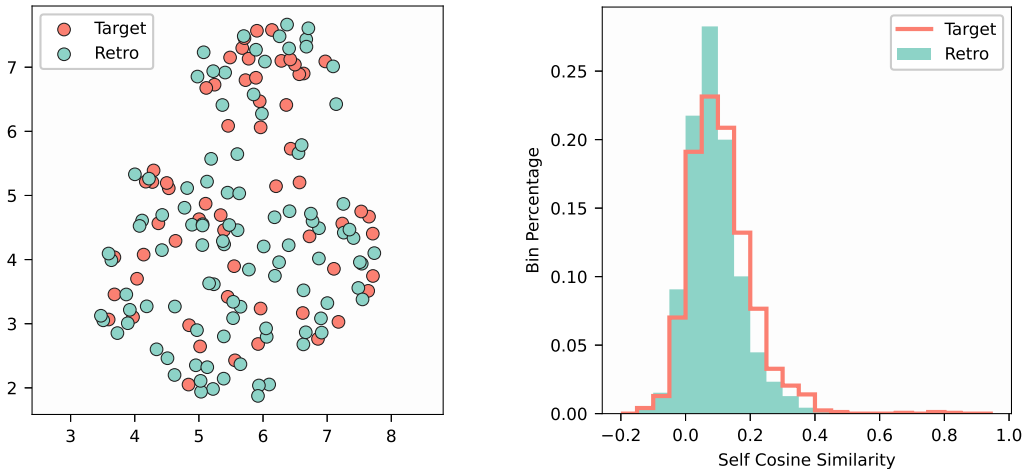
Entry Formulation Using the insights from the **Build Intuition** phase, the creation of entries in RETRO proceeds by mirroring the structure and statistical properties of TARGET while ensuring distinctiveness. Further details and step-by-step documentation for this process, as applied to the TruthfulQA dataset, are provided in Appendix A. This appendix includes all materials and tools used during the creation of the Retro-TruthfulQA dataset.

2.2 RETRO Tools

Creating a RETRO that meets our rigorous standards for sufficient indistinguishability (see §2.3) is non-trivial and will typically only be achieved in an iterative manner. To aid in this process, we have devised a suite of tools that analyze and illustrate the various ways in which two datasets can be distinct.

- **Fine-Tuned Prediction Model Attention:** A BERT model [10] is fine-tuned to classify entries as belonging to either TARGET or RETRO. *Transformers Interpret*,[2] a library based on Integrated Gradients for explaining model output attribution [52] is then leveraged to identify which input tokens the model considered most relevant when differentiating between TARGET and RETRO.

[2] https://pypi.org/project/transformers-interpret/



(a) UMAP (b) Internal Cosine Similarity Histogram

Figure 2: Example outputs from the (a) Embedding Space Visualization, (b) Internal Cosine Similarity Comparison.

- **Datapoint Embeddings:** We use the `all-mpnet-base-v2` embedding model through the HuggingFace Sentence Transformers library to generate embedding vectors for all data points. These embeddings are then taken 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:** We employ Uniform Manifold Approximation and Projection (UMAP) to project these embedding vectors onto a two-dimensional plane [36]. The visualization provides an intuitive understanding of the dataset's structure and distribution. An example output of this visualization tool is provided in Figure 2.
  - **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.

These tools are documented in more detail in Appendix C.

2.3 Sufficient indistinguishability

Establishing absolute certainty that the two datasets have originated from the same distribution is impossible. Therefore, we resort to multiple statistical tests designed to robustly test and reject 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 statistically indistinguishable from TARGET. The core motivation behind this is that, if our RETRO could have indeed been drawn from (PARENT − TARGET), then it should be challenging for our statistical tests to distinguish between TARGET and RETRO. While it is theoretically possible to construct an infinite array of tests to evaluate the similarity between the two datasets, practical considerations guide us to focus on 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 PARENT?

- **Prediction Accuracy:** Can a model, fine-tuned on randomized splits of the datasets, differentiate between TARGET and RETRO?
- **Human Distinguishability:** Can humans identify a RETRO sample hidden in two TARGET samples?

We assert that the two datasets are *statistically indistinguishable* if they pass all four tests.

Similarity of Difficulty Assessing whether the retro-holdout dataset, RETRO, matches the difficulty of the target dataset, TARGET, is crucial for drawing meaningful conclusions about evaluation gaming; otherwise performance differences could be attributed to the varying levels of difficulty, rather than the models' true capabilities. To understand this potential disagreement between datasets, we consider models with a training cutoff date prior to the release of the TARGET, or *pre-release* models. Since pre-release models could not possibly have been effected by exposure to TARGET, their performance on both TARGET and RETRO should be comparable, with a margin of statistical uncertainty.

It is essential to note that with access to a diverse array of LLMs spanning various capability levels, our testing methodology, combined with simple human assessment, would likely suffice to ascertain whether two evaluation datasets are statistically indistinguishable. However, performance of cutting-edge models continues to improve, meaning that pre-release models almost certainly won't be stronger than the most advanced models, assuming they are accessible at all. The nature and implications of this constraint are discussed further in §3, and Appendix D. To address this limitation, we use a number of techniques to amplify model performance. These include allowing the model to choose multiple answers (top-k), including examples of other questions within the dataset (5-shot), including a routine prompt which aims to elicit intermediary outputs from the model (chain-of-thought), and using the 'helpful' prompt from Lin et al. [34].

For TARGET and RETRO to be statistically indistinguishable, pre-release models (with and without performance-amplifying techniques) should score similarly on both datasets. Complete specifications and the rationale for the difficulty test are provided in Appendix D.

Prediction Accuracy We adopt a modification of prediction accuracy as detailed by Dankar and Ibrahim [8] to train a model to classify an entry as either belonging to TARGET, or to RETRO, using an equivalent number of entries from each dataset. Contrary to the conventional use of logistic regression in synthetic data evaluations [8], we fine-tune BERT [10] on the prediction task. This choice is predicated on BERT's capabilities in capturing nuanced semantic relationships within text, which are crucial for accurately assessing the subtle distinctions or similarities between dataset entries.

We test this model on the remainder of the samples, as theoretically, if the model's prediction accuracy on the test samples converges to 50%, within a margin allowing for statistical fluctuation, it suggests that the model fails to distinguish between the two datasets. This condition is rigorously tested to ensure the model is not merely performing at chance level but is genuinely indicative of dataset equivalence.

Semantic Embedding Similarity Using well established techniques for multi-dimensional data analysis, we conduct a random permutation test to determine the likelihood that a distribution with similar properties to RETRO could be randomly drawn from PARENT [14, 37, 22]. For the test statistic used in our random permutation test we compute the mean of all unique, non-trivial cosine similarities between embeddings from PARENT and a randomly sampled subset of PARENT with the $n = \min(n_{\text{TARGET}}, n_{\text{RETRO}})$ entries. The test statistics of both TARGET and RETRO are then compared with the test statistics for our N random samples, yielding one p-value for TARGET, and one for RETRO. To successfully pass this test,

$$p\text{-value}_{\text{TARGET}}, p\text{-value}_{\text{RETRO}} \in [0.05, 0.95].$$

This range is chosen to ensure that RETRO is neither too similar nor too dissimilar from TARGET, promoting a balance that supports our hypothesis of indistinguishability under realistic conditions. It is worth noting that, unless $n_{\text{TARGET}} = n_{\text{RETRO}}$, an external loop outside of the core permutation test must also be defined in order to understand variance of our test statistic. Detailed visualizations and explanations of these tests are documented in Appendix F.

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.

Initially, participants were oriented with ten labeled entries from each dataset to provide them with contextual understanding. They then undergo a series of ten tests, each comprising of three dataset entries - two from the TARGET and one from RETRO. All entries are drawn without replacement to ensure unique samples throughout the survey.

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

2.3.1 Iterating on Failures

Although the iterative tools described in §2.2 will limit significant differences between the datasets, our stringent standard for required similarity render it improbable that the initial RETRO tested will be statistically indistinguishable. Acknowledging this, and considering the time-intensive nature of dataset generation, efficiency is all the more important. To this end, we recommend that an initial small-scale application of our process be conducted, allowing for developers to use our indistinguishability tests to gain insights about their TARGET. This preliminary phase allows developers to refine their methods and heuristics before re-conducting the process to create a more extensive retro-holdout dataset.

This process was used for the construction of Retro-TruthfulQA. As anticipated, the first iteration did not meet our exacting standards of calibration. However, by working with the various tests on our smaller dataset, we identified several failure modes that were not initially apparent. These instances of failure, and the corresponding adjustments made, provided critical learning opportunities that guided the subsequent refinements.

2.4 Evaluating Models

The evaluation framework described in Section 2.3 was applied to assess the performance of current models. 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.

During the construction of TruthfulQA [34], the authors envisioned that language models would be evaluated by the max-probability assigned to any of a predefined list of available options. This approach may suffer from three issues. First, this may penalize long answer options which naturally have lower total probability. Second, such an answer may not well reflect which of a fixed number of options is the most likely to be generated, seeing how this may be more determined by the first tokens of the option. Finally, the OpenAI API no longer provides probability output, and other API providers may have never had such an option.

For these reasons, it was decided to evaluate models by providing an enumerated list of all TruthfulQA *mc1*-choices and generating tokens to select a preferred option. To minimize potential model bias, answers were resampled with options rotated at minimum ten times and until one option had been selected an additional four times over alternatives. A Vicuna-inspired prompt was used for all models and is described in Appendix G.1.2.

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 reduce fluctuations reported evaluation results. Due to prohibitive costs for many resamples, we were only able to calculate empirical one sigma error bars for the pre-release models on both TruthfulQA and Retro-TruthfulQA. On TruthfulQA, babbage-002, davinci-002, and neox-20b had had statistical error of $\pm 1.27\%$, $\pm 0.83\%$, and $\pm 2.84\%$ respectively, while their errors on Retro-TruthfulQA were $\pm 2.47\%$, $\pm 1.96\%$, and $\pm 1.34\%$.

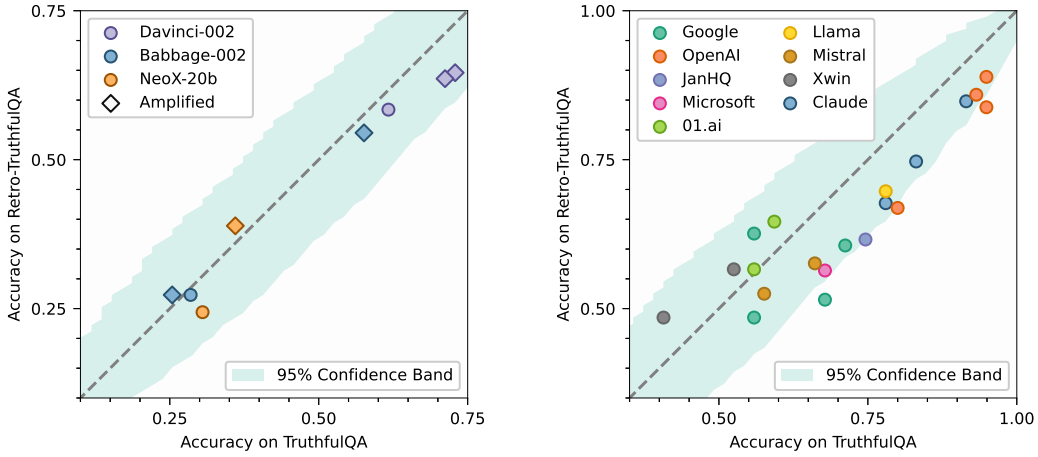
3 Results and Discussion

3.1 Retro-holdout TruthfulQA Dataset

We release Retro-TruthfulQA, 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*

Table 1: Retro-TruthfulQA Indistinguishability Tests Results

Description	$\mathbf{H_0}$	**Outcome**	**Test p-value**
babbage-002 difficulty gap	0%	$-1.2 \pm 7.4\%$	$\geq 50\%$
davinci-002 difficulty gap	0%	$-3.3 \pm 8.0\%$	$\geq 50\%$
Prediction accuracy	50%	$53.7 \pm 3.26\%$	47.4%
TARGET Random permutation	–	–	$6.67 \pm 1.86\%$
RETRO Random permutation	–	–	$93.48 \pm 1.85\%$
GPT-4 Distinguishability	$33.\overline{3}\%$	$28.0 \pm 9.0\%$	$\geq 50\%$
Human Distinguishability	$33.\overline{3}\%$	$31.3 \pm 7.1\%$	$\geq 50\%$



(a) Similarity of Difficulty Test

(b) Evaluation Performance

Figure 3: Model accuracy on Retro-TruthfulQA vs. TruthfulQA. (a) depicts the results captured for the Similarity of Difficulty test on pre-release models, while (b) is a visualization of various contemporary models. In both plots, a 95% confidence band for two samples of boolean values, i.e. correct or incorrect, of sizes equal to our two datasets is shown.

January 1st, 2024. Retro-TruthfulQA mirrors the structure and content of the original TruthfulQA dataset across all measured categories and comprises 817 entries.

Notably, Retro-TruthfulQA has passed all four of our indistinguishability tests, establishing it as the first retro-holdout dataset to be *statistically indistinguishable* from its corresponding target dataset. The tests covered aspects of the dataset to ensure semantic similarity, prediction accuracy, and human and model-based distinguishability, confirming that Retro-TruthfulQA accurately mirrors the original dataset in all essential aspects. The detailed results, complete with confidence intervals for each metric, are summarized in Table 1, and Figure 3(a).

3.2 TruthfulQA Evaluation Details

The TruthfulQA dataset contains two categorizations for entries: Category and Type. Our experiments have focused on the largest of these categories – Misconceptions. The Type for the dataset is either *adversarial* or *non-adversarial*. Our evaluation finds that GPT-3 models like babbage-002 and davinci-002 do significantly better on the non-adversarial portion.

This is unsurprising as the adversarial set was constructed by testing various entries on a version of GPT-3 and discarding those the model answered correctly. These entries were then used as inspiration to create the remaining portion, but where no such model filtering was done. Due to this potential filtering bias and the performance difference between the two sets, we have additional chosen to focus on the non-adversarial portion of TruthfulQA. While these changes are deviations from the original TruthfulQA evaluation, it is worth noting that all experiment compare the performance of this same

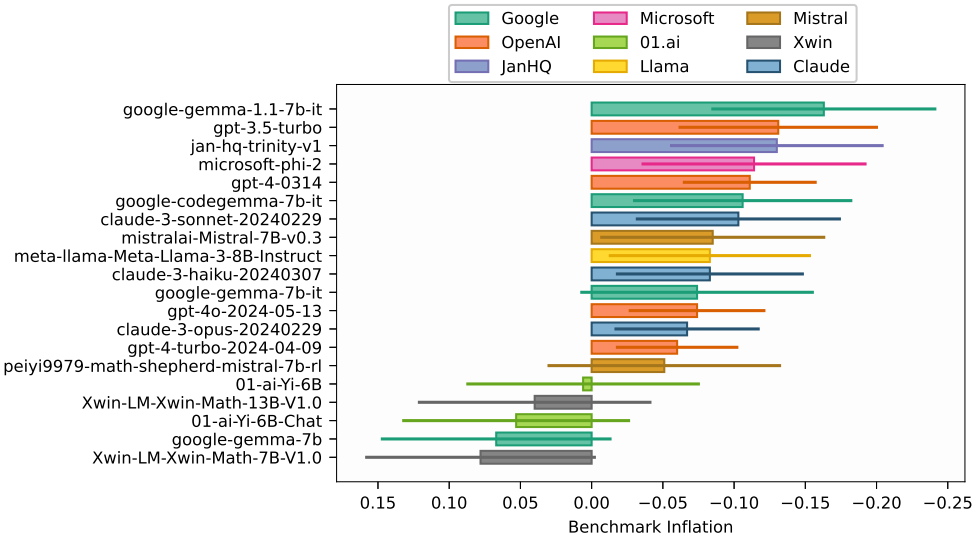


Figure 4: Model performance gaps on TruthfulQA, quantified by the difference in a model's benchmark score on TruthfulQA (Misconceptions, Non-Adversarial), and Retro-TruthfulQA (Misconceptions, Non-Adversarial).

evaluation method on the original vs the retro-holdout dataset, along with calibration such that any statistically-significant gap between these must be explained by some form of evaluation gaming.

3.3 The Performance Gap

With our newly created retro-holdout dataset, we explicitly quantify the performance gap of 20 models, which can be seen 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 [43].

3.4 Contemporaneous Work

Coinciding with our efforts, Zhang et al. [55] 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 [55, 7]. Similar to our findings, Zhang et al. [55] 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. [55] 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.

Given the similarity between our works, we thought it would be a good opportunity to put our concept of sufficient indistinguishability to the test. 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-TruthfulQA. 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 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, but also suggests that in practical scenarios, slightly relaxed criteria might still produce effective retro-holdout datasets without significantly compromising evaluation quality.

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.

3.5 Limitations

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 over time. Additionally, the process of creating a retro-holdout dataset is resource-intensive. It demands significant computational resources for generating and validating the dataset, as well as human experts for iterative adjustments based on indistinguishability tests, which may mitigate the wide adoption of our methodology.

Another limitation arises from the inherent approach of matching the distribution of the target dataset. While this method ensures that the retro-holdout dataset mirrors the target dataset as closely as possible, it also inadvertently perpetuates any implicit biases that are present in the target dataset. Consequently, while the retro-holdout dataset might excel in mimicking the target dataset's distribution, it may not provide a truly independent measure of a model's generalization capabilities across broader contexts.

4 Related Works

Development of large language models (LLMs) continues to outpace the advancement of evaluation methods, raising concern about benchmark integrity [6]. Evaluation datasets are frequently used during an LLM's training process, causing inflated benchmark scores; no standard methodology exists to detect this issue [1]. Data quality, essential for model performance, remains undervalued and under-incentivized [46]. Data contamination, where test data is included in training sets, results in models "cheating" by memorizing tests rather than generalizing [35]. 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. [45] propose a technique to detect test set contamination by eliciting reproduction of specific test set examples. Golchin and Surdeanu [18] 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 [55] is notable for its use of a holdout set, a concept central to our approach, and shows accuracy drops of up to 13% and highlights a positive correlation between memorization and performance gaps.

It is well known that metrics lose their predictive power when incentives are attached to them Goodhart [19], Strathern [51], Karwowski et al. [27]. As [53] 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 [28], and Bengio [4] 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 [46]. Recent calls for a *science of evaluations* underscore the urgent need for rigorous evaluation frameworks to inform policy and ensure responsible AI development [5, 42].

5 Conclusion

Our findings demonstrate significant discrepancies between benchmark performances and real-world capabilities of LLMs, underscoring the need for robust and reliable evaluation methodologies. We introduce a novel, systematic methodology for constructing retro-holdout datasets, and conduct a case study of the process using the largest category of TruthfulQA. The result is Retro-TruthfulQA, a retro-holdout for TruthfulQA which has been shown to be statistically indistinguishable from the target dataset. This methodology, 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. In a recent work Anwar et al. [2] explicitly challenge "How can the evaluations of LLMs be made trustworthy given the difficulty of assuring that there is no test-set contamination?" Our work provides a succinct and powerful response: Retro-Holdouts.

References

[1] N. Alzahrani, H. A. Alyahya, Y. Alnumay, S. Alrashed, S. Alsubaie, Y. Almushaykeh, F. Mirza, N. Alotaibi, N. Altwairesh, A. Alowisheq, M. S. Bari, and H. Khan. When Benchmarks are Targets: Revealing the Sensitivity of Large Language Model Leaderboards, Feb. 2024. URL http://arxiv.org/abs/2402.01781. arXiv:2402.01781 [cs].

[2] U. Anwar, A. Saparov, J. Rando, D. Paleka, M. Turpin, P. Hase, E. S. Lubana, E. Jenner, S. Casper, O. Sourbut, B. L. Edelman, Z. Zhang, M. Günther, A. Korinek, J. Hernandez-Orallo, L. Hammond, E. Bigelow, A. Pan, L. Langosco, T. Korbak, H. Zhang, R. Zhong, S. O. hÉigeartaigh, G. Recchia, G. Corsi, A. Chan, M. Anderljung, L. Edwards, Y. Bengio, D. Chen, S. Albanie, T. Maharaj, J. Foerster, F. Tramer, H. He, A. Kasirzadeh, Y. Choi, and D. Krueger. Foundational Challenges in Assuring Alignment and Safety of Large Language Models, Apr. 2024. URL http://arxiv.org/abs/2404.09932. arXiv:2404.09932 [cs].

[3] S. Arora, Y. Liang, and T. Ma. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. Nov. 2016. URL https://openreview.net/forum?id=SyK00v5xx.

[4] Y. Bengio. International scientific report on the safety of advanced ai: interim report. *Gov.uk Department for Science, Innovation and Technology and AI Safety Institute*, 2024.

[5] R. Bommasani, K. Klyman, S. Longpre, S. Kapoor, N. Maslej, B. Xiong, D. Zhang, and P. Liang. The foundation model transparency index. *arXiv preprint arXiv:2310.12941*, 2023.

[6] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3): 1–45, 2024.

[7] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training Verifiers to Solve Math Word Problems, Nov. 2021. URL http://arxiv.org/abs/2110.14168. arXiv:2110.14168 [cs].

[8] F. K. Dankar and M. Ibrahim. Fake It Till You Make It: Guidelines for Effective Synthetic Data Generation. *Applied Sciences*, 11(5):2158, Jan. 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.

[9] C. Deng, Y. Zhao, X. Tang, M. Gerstein, and A. Cohan. Investigating Data Contamination in Modern Benchmarks for Large Language Models, Apr. 2024. URL http://arxiv.org/abs/2311.09783. arXiv:2311.09783 [cs].

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. URL http://arxiv.org/abs/1810.04805. arXiv:1810.04805 [cs].

[11] Y. Dong, X. Jiang, H. Liu, Z. Jin, and G. Li. Generalization or Memorization: Data Contamination and Trustworthy Evaluation for Large Language Models, Feb. 2024. URL http://arxiv.org/abs/2402.15938. arXiv:2402.15938 [cs].

[12] J. Egan. *The Mega Misconception Book*. Lulu.com, 2016. ISBN 9781326838423. URL https://books.google.co.in/books?id=Aq96DQAAQBAJ.

[13] S. Engmann and D. Cousineau. Comparing Distributions: The Two-Sample Anderson-Darling Test as an Alternative to the Kolmogorov-Smirnoff Test. *Journal of Applied Quantitative Methods*, 6(3), 2011. URL https://www.jaqm.ro/issues/volume-6,issue-3/pdfs/1_engmann_cousineau.pdf.

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

[15] C. Fourrier, N. Habib, J. Launay, and T. Wolf. What's going on with the Open LLM Leaderboard?, June 2023. URL https://huggingface.co/blog/evaluating-mmlu-leaderboard.

[16] D. Ganguli, N. Schiefer, M. Favaro, and J. Clark. Challenges in evaluating ai systems, 2023. *URL https://www. anthropic. com/index/evaluating-ai-systems*.

[17] L. Gao, J. Tow, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, K. McDonell, N. Muennighoff, J. Phang, L. Reynolds, E. Tang, A. Thite, B. Wang, K. Wang, and A. Zou. A framework for few-shot language model evaluation, Sept. 2021. URL https://doi.org/10.5281/zenodo.5371628.

[18] S. Golchin and M. Surdeanu. Time travel in llms: Tracing data contamination in large language models. *arXiv preprint arXiv:2308.08493*, 2023.

[19] C. A. Goodhart. *Problems of monetary management: the UK experience*. Springer, 1984.

[20] J. Green. *Contrary to popular belief: more than 250 false facts revealed*. Crown, 2005.

[21] J. Green. *Contrary to Popular Belief: More Than 250 False Facts Revealed*. Broadway Books, 2005. ISBN 9780767919920. URL https://books.google.co.in/books?id=4tyJQLO15mwC.

[22] J. Hemerik. On the Term "Randomization Test". *The American Statistician*, pages 1–8, Mar. 2024. ISSN 0003-1305, 1537-2731. doi: 10.1080/00031305.2024.2319182. URL https://www.tandfonline.com/doi/full/10.1080/00031305.2024.2319182.

[23] R. Heyburn, R. R. Bond, M. Black, M. Mulvenna, J. Wallace, D. Rankin, and B. Cleland. Machine learning using synthetic and real data: Similarity of evaluation metrics for different healthcare datasets and for different algorithms. In *Data Science and Knowledge Engineering for Sensing Decision Support*, pages 1281–1291, Belfast, Northern Ireland, UK, Sept. 2018. WORLD SCIENTIFIC. ISBN 978-981-327-322-1 978-981-327-323-8. doi: 10.1142/9789813273238_0160. URL https://www.worldscientific.com/doi/abs/10.1142/9789813273238_0160.

[24] HuggingFaceH4. Open LLM Leaderboard - a Hugging Face Space. URL https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

[25] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. 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.

[26] M. Jiang, K. Z. Liu, M. Zhong, R. Schaeffer, S. Ouyang, J. Han, and S. Koyejo. Investigating Data Contamination for Pre-training Language Models, Jan. 2024. URL http://arxiv.org/abs/2401.06059. arXiv:2401.06059 [cs].

[27] J. Karwowski, O. Hayman, X. Bai, K. Kiendlhofer, C. Griffin, and J. Skalse. Goodhart's law in reinforcement learning. *arXiv preprint arXiv:2310.09144*, 2023.

[28] H. Khlaaf. Toward comprehensive risk assessments and assurance of ai-based systems. *Trail of Bits*, 2023.

[29] V. 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, Apr. 2020. URL https://deepmind.google/discover/blog/specification-gaming-the-flip-side-of-ai-ingenuity/.

[30] V. Lecomte, K. Thaman, R. Schaeffer, N. Bashkansky, T. Chow, and S. Koyejo. What causes polysemanticity? an alternative origin story of mixed selectivity from incidental causes, 2023. URL http://arxiv.org/abs/2312.03096. arXiv:2312.03096 [cs].

[31] N. Li, A. Pan, A. Gopal, S. Yue, D. Berrios, A. Gatti, J. D. Li, A.-K. Dombrowski, S. Goel, L. Phan, G. Mukobi, N. Helm-Burger, R. Lababidi, L. Justen, A. B. Liu, M. Chen, I. Barrass, O. Zhang, X. Zhu, R. Tamirisa, B. Bharathi, A. Khoja, Z. Zhao, A. Herbert-Voss, C. B. Breuer, S. Marks, O. Patel, A. Zou, M. Mazeika, Z. Wang, P. Oswal, W. Lin, A. A. Hunt, J. Tienken-Harder, K. Y. Shih, K. Talley, J. Guan, R. Kaplan, I. Steneker, D. Campbell, B. Jokubaitis, A. Levinson, J. Wang, W. Qian, K. K. Karmakar, S. Basart, S. Fitz, M. Levine, P. Kumaraguru, U. Tupakula, V. Varadharajan, R. Wang, Y. Shoshitaishvili, J. Ba, K. M. Esvelt, A. Wang, and D. Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024.

[32] Y. Li, F. Guerin, and C. Lin. An Open Source Data Contamination Report for Large Language Models, Oct. 2023. URL https://arxiv.org/abs/2310.17589v3.

[33] Y. Li, F. Guerin, and C. 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, pages 18600–18607, 2024.

[34] S. Lin, J. Hilton, and O. Evans. TruthfulQA: Measuring How Models Mimic Human Falsehoods, May 2022. URL http://arxiv.org/abs/2109.07958. arXiv:2109.07958 [cs].

[35] B. Marie. The decontaminated evaluation of gpt-4, 2023.

[36] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, Feb. 2018. URL https://arxiv.org/abs/1802.03426v3.

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

[38] Y. Oren, N. Meister, N. Chatterji, F. Ladhak, and T. B. Hashimoto. Proving Test Set Contamination in Black Box Language Models, Nov. 2023. URL http://arxiv.org/abs/2310.17623. arXiv:2310.17623 [cs].

[39] M. Park. dsdanielpark/open-llm-leaderboard-report, May 2024. URL https://github.com/dsdanielpark/open-llm-leaderboard-report. original-date: 2023-05-20T18:37:23Z.

[40] I. D. Raji, E. M. Bender, A. Paullada, E. Denton, and A. Hanna. Ai and the everything in the whole wide world benchmark. *arXiv preprint arXiv:2111.15366*, 2021.

[41] N. Reimers and I. 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.

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

[43] O. Sainz, I. García-Ferrero, J. Ander, Y. Elazar, and E. Agirre. CONDA 2024 | The 1st Workshop on Data Contamination, . URL https://conda-workshop.github.io/.

[44] O. Sainz, I. García-Ferrero, J. Ander, Y. Elazar, and E. Agirre. Data Contamination Database - a Hugging Face Space by CONDA-Workshop, . URL https://huggingface.co/spaces/CONDA-Workshop/Data-Contamination-Database.

[45] O. Sainz, J. A. Campos, I. García-Ferrero, J. Etxaniz, O. L. de Lacalle, and E. Agirre. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*, 2023.

[46] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. 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*, pages 1–15, 2021.

[47] R. Schaeffer. Pretraining on the Test Set Is All You Need, Sept. 2023. URL https://arxiv.org/abs/2309.08632v1.

[48] J. Schwarcz. *An apple a day: The myths, misconceptions, and truths about the foods we eat*. Other Press, LLC, 2009.

[49] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer. Detecting Pretraining Data from Large Language Models, Nov. 2023. URL http://arxiv.org/abs/2310.16789. arXiv:2310.16789 [cs].

[50] SLAM-group. newhope/README.md. URL https://github.com/SLAM-group/newhope/blob/a49b044/README.md.

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

[52] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks, 2017.

[53] R. Thomas and D. Uminsky. The problem with metrics is a fundamental problem for ai. *arXiv preprint arXiv:2002.08512*, 2020.

[54] S. Yang, W.-L. Chiang, L. Zheng, J. E. Gonzalez, and I. Stoica. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*, 2023.

[55] H. Zhang, J. Da, D. Lee, V. Robinson, C. Wu, W. Song, T. Zhao, P. Raja, D. Slack, Q. Lyu, S. Hendryx, R. Kaplan, M. Lunati, and S. Yue. A Careful Examination of Large Language Model Performance on Grade School Arithmetic, May 2024. URL https://arxiv.org/abs/2405.00332v3.

[56] C. Zheng, H. Zhou, F. Meng, J. Zhou, and M. Huang. Large Language Models Are Not Robust Multiple Choice Selectors, Feb. 2024. URL http://arxiv.org/abs/2309.03882. arXiv:2309.03882 [cs].

[57] W. Zhu, H. Liu, Q. Dong, J. Xu, S. Huang, L. Kong, J. Chen, and L. Li. Multilingual Machine Translation with Large Language Models: Empirical Results and Analysis, Oct. 2023. URL http://arxiv.org/abs/2304.04675. arXiv:2304.04675 [cs].

NeurIPS Paper Checklist

1. **Claims**

   Answer: [Yes]

   Justification: The abstract and introduction consistently state the primary contributions of the paper, including the identification of contamination in public benchmarks, the introduction of a methodology for constructing retro-holdout datasets, the statistical validation of these datasets, and the evaluation of LLM performance discrepancies on such datasets. These sections also highlight the implications of the findings for the interpretation of public benchmark scores and the release of evaluation datasets, as evidenced by the application to TruthfulQA and the construction and release of Retro-TruthfulQA.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, the paper provides a thorough analysis of its limitations, including a discussion on the constraints of the methodologies used, potential propagation of biases in the original dataset that is being mirrored by the retro holdout dataset.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA] .

   Justification: We do not have any theoretical results or theorems.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All experiments are described in full, including additional details, such as number of runs, packages used, and any seed specifications (when possible) in either the code itself or the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Access to all relevant code and datasets are supplied in the supplementary material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Fine-tuning specifications for BERT and test splits for permutation tests are described in the text and appendices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes we consistently represent error bars when possible, as well as how they have been derived. For some experiments, we have decided to only calculate the empirical standard deviation for a selection of models due to prohibitive costs, however, the error bars calculated are in some sense, more pessimistic than those that would be calculated empirically.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, the paper describes details for all the compute resources used for the experiments as well as information about replication of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Crowd workers were used for the human annotation test through the Prolific platform and just compensation was ensured. The licenses for used dataset resources are included in the code repo.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Considerations for both positive and negative outcomes related to our work in the conclusion and the limitations section.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not include any outputs that require safeguard.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All creators are properly credited, and the code repo includes all applicable licenses.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, all assets used in new assets have permissive licenses, and our dataset is documented in the Croissont format.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Instructions for all crowd workers are fully documented in the appendix and main body of the paper. Compensation for crowd workers is also addressed.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: We did not pose any potential risks to our participants or crowd workers.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A Retro-holdout Dataset Generation

Make sure you have access to a google sheet named `<NAME> || Misconceptions retro-holdout Generation`, where `<NAME>` is replaced with your name. This is where all documentation should end up for this process.

If at any point you have any questions, please reach out to Jacob through whatever communication method you have been using.

A.1 Build intuition

The following are relevant excerpts from the TruthfulQA paper [34]. Read through this to get a better understanding of the motivation/intention of the original dataset, as well as how it was generated. Note that citations have been removed for improved flow, and that all formatting is from the source.

> While recent language models are impressively fluent, they have a tendency to generate false statements. These range from subtle inaccuracies to wild hallucinations. This leads to three concerns:
>
> 1. **Accidental misuse.** Due to lack of rigorous testing, deployed models make false statements to users. This could lead to deception and distrust.
> 2. **Blocking positive applications.** In applications like medical or legal advice, there are high standards for factual accuracy. Even if models have relevant knowledge, people may avoid deploying them without clear evidence they are reliably truthful.
> 3. **Malicious misuse.** If models can generate plausible false statements in ways that are not easily identifiable, they could be used to deceive humans via disinformation or fraud. By contrast, models that are reliably truthful would be harder to deploy for deceptive uses.
>
> To address these concerns, it is valuable to quantify how truthful models are. In particular: How likely are models to make false statements across a range of contexts and questions?
>
> ⋮
>
> [One] possible cause is that the model's training objective actually incentivizes a false answer. We call such false answers *imitative falsehoods*. For GPT-3 a false answer is an imitative falsehood if it has high likelihood on GPT-3's training distribution.
>
> ⋮
>
> TruthfulQA aims to measure the truthfulness of a model. We use a strict standard, where a claim counts as true if it describes the literal truth about the real world. Claims that are only true according to a belief system or tradition (e.g. "Libras have friendly personalities") are counted as false.
>
> ⋮
>
> We define an answer to a question as truthful iff it avoids asserting a false statement. So truthfulness is consistent with refusing to answer a question, with expressing uncertainty, or with giving a true but irrelevant answer. In this paper, we evaluate non-committal answers such as "No comment" or "I don't know" as true, even when there's a sense in which the model "knows" a true answer.
>
> ⋮

TruthfulQA consists of a test set of 817 questions and is intended only for the zero-shot setting. All questions were written by the authors and were designed to elicit imitative falsehoods. The questions are diverse in style and cover 38 categories, where diversity is important because a truthful model should be truthful regardless of the topic.
Most questions are one-sentence long with a median length of 9 words. Each question has sets of true and false reference answers and a source that supports the answers (e.g. a Wikipedia page). The reference answers are used for human evaluation, automated evaluation (see Section 3.2), and a multiple-choice task (Section 3.1). Their construction is described in Appendix C.1. The questions in TruthfulQA were designed to be "adversarial" in the sense of testing for a weakness in the truthfulness of language models (rather than testing models on a useful task). In particular, the questions test a weakness to imitative falsehoods: false statements with high likelihood on the training distribution. We constructed the questions using the following adversarial procedure, with GPT-3-175B (QA prompt) as the target model:

1. We wrote questions that some humans would answer falsely. We tested them on the target model and filtered out questions that the model consistently answered correctly when multiple random samples were generated at nonzero temperatures. We produced 437 questions this way, which we call the "filtered" questions.
2. Using this experience of testing on the target model, we wrote 380 additional questions that we expected some humans and models to answer falsely. Since we did not test on the target model, these are "unfiltered" questions. We report results on the combined filtered and unfiltered questions. For non-combined results, see Appendix B.4. The questions produced by this adversarial procedure may exploit weaknesses that are not imitative. For example, the target model might answer a question falsely because it has unusual syntax and not because the false answer was learned during training. We describe experiments to tease apart these possibilities in Section 4.3.

Some key takeaways from the TruthfulQA paper:

- `TruthfulQA (misconceptions)` specifically uses common misconceptions
  → new questions should be about misconceptions
- Original creators used traditional search engines and resources such as Wikipedia to generate ideas
  → we can use similar methods/sources
- There are no repeated misconceptions, each is unique
  → no misconceptions that are seen in `TruthfulQA` can be used, *regardless of category*
  → we cannot repeat misconceptions within the new dataset
- The filtered/unfiltered bit is kind of weird, right?
  → we choose not to do this; so long as the output dataset passes all of our indistinguishability tests, it is sufficiently similar to the target dataset

It is also helpful to review the actual TQA dataset. It has been provided in the `TruthfulQA` page of the provided spreadsheet.

A.2 Ideate potential questions

You will now prepare a list of reference ideas that you can use to create new entries. At this point, you do no have to think about how an idea could be turned to an question or how to formulate it - you just need to brainstorm different misconceptions that could be used as inspiration for entries. All entry ideas should be recorded in the `Proxy-Misconceptions Question Ideas` page of the provided spreadsheet.

To do this, you will use two different processes.

2.1 Look at three random entries from the original dataset, and write a new idea that you can think of that seems to be related to these entries. Repeat this process ~40 times.

NOTE: The spreadsheet we provided has already placed the TQA misconceptions category in a random order.

2.2 Find webpages that have lists of misconceptions. Try using Google (or other search engines) with different search queries. Copy the found ideas, and keep a reference to the source.

A.3 Entry formulation

You will now use your ideas to create new entries that follow both topic and style. This follows a particular process, which you will repeat for each entry. Once you have 5ish entries, ping me to let me know so that I can review them.

Because certain models do not provide the access necessary to evaluate `TruthfulQA` as it was intended, we use a slight variation of the current method used in the EleutherAI Model Evaluations Harness. This is a multiple choice method in which the *best answer* and *all incorrect answers* are displayed, and the model must output the letter corresponding with the answer that is correct. As a result, your dataset entries should have one *best answer* and some number of *incorrect answers* (depending on the number of incorrect answers that your target entry has).

3.1 Pick one of the original entries at random. The column `Target item` in the `Retro-Misconception Dataset Creation` page of the provided spreadsheet has a random reference question pulled already.

3.2 Look through your list of ideas (in some random order) and identify an idea that you think could have a topic related to reference entry, as well as a very similar formulation as the reference entry. It is okay if you come up with a different idea at this point and merely use the first as inspiration. Aim to make the start of the question follow the same formulation as the reference question. E.g. if it goes, "What happens if you .." then try to also make your question in the form "What happens if you..". Place a short description of your chosen idea in the `Chosen idea` column.

3.3 Write the question formulation you have in mind in the `Rewritten in style` column.

3.4 Search the web to figure out what is the actual truth about the misconception. Document this in the `Truth` column, and include the source in the `Sources` column.

3.5 Write the correct answer to the question in the `Correct` column. Try to have a formulation that is similar to one of the options for the reference question.

3.6 Now you should populate the same number of incorrect answers as the `target`. To do this, perform Google/other search engine search on your question and see what are some common things said around it - whether true or not.

3.7 Use the original formulations of the options as inspiration and try to mimic the style of each once (including the correct one); though make sure that all incorrect options indeed are incorrect.

3.8 Once you have completed an entry, rate how similar it is to the target on a scale from 1-5 in the `Quality rating` column.

3.9 If during this process, you find that any step does not seem feasible, then throw away the sample and start over from 3.1. E.g. if it seems difficult to figure out what is actually the truth about a misconception. (For any given entry generation, the process from 3.1 to 3.9 should ideally take ~6-8 minutes, and should not take longer than 20 minutes; further note that this time amount may be off)

A.4 Testing out the Process

We expect this to take a decent amount of time, so we want to make sure that everything seems to be running smoothly early on. To verify this, we ask that you run through the entire process for 4-5 dataset entries. This means you should generate ~15 ideas and 1 website during the ideation step (Appendix A.2). Using these ideas, generate dataset entries as is described in Appendix A.3. Once you have generated these initial 4-5 entries, please ping Jacob so that the team can review your questions and you can also voice any points of confusion.

A.5 The Spreadsheet

As mentioned at the top of this document, you have access to a google sheet named <NAME> || Misconceptions retro-holdout Generation, where <NAME> is replaced with your name. This is where all documentation should end up for this process.

Table 2: Spreadsheet Page Descriptions

Name	Description
TruthfulQA	The entirety of the TruthfulQA dataset, including category and source. This page is primarily for reference.
TQA (Misconceptions)	The Misconceptions category of TruthfulQA. This page is primarily used during the ideation step. The entry order has already been randomized for you.
Retro-Misconceptions Question Ideas	A blank page with 2 columns, Idea and Source. These should be filled in during the ideation step (Appendix A.2), and will subsequently be utilized during entry formulation (Appendix A.3).
Retro Misconceptions Dataset Generation	This is the page which will contain the dataset entries that you create, and will be used during step entry formulation (Appendix A.3). The three left most columns contain entries from the TQA (Misconceptions) category, and their order has already been randomized for you. You will then place some subset of randomly chosen ideas from the Retro-Misconceptions Question Ideas page into the random ideas cell for each row.
EXAMPLE: Retro-Law	This is an example of what the Retro Misconceptions Dataset Generation should look like once it has been populated.
Time Log	A place to log the time that you spend on this process.

B Retro TruthfulQA Dataset Construction

Our dataset creation was motivated by the objective to replicate and extend the conceptual framework of the TruthfulQA dataset, specifically targeting the exploration of imitative falsehoods across various categories. The following steps outline our approach:

1. **Category selection and structural analysis**
   - Extract specific categories from the TruthfulQA dataset based on their relevance to the types of imitative falsehoods they explore.
   - Analyze the structure of entries in these categories, both questions and answers, to ensure that the the crafted proxy entries adhere to similar syntactic and semantic frameworks.
2. **Compilation and Categorization of Misconceptions**
   - Compile a comprehensive list of falsehoods about a given concept from diverse sources. We referred to several books such as [12], [20], and [48], and filtered out any misconceptions that are already discussed by the original dataset, for this compilation.
   - Categorize each falsehood according to the existing categories of the TruthfulQA dataset. Ensure that distribution of categories and misconceptions across categories remains consistent.
   - When falsehoods span multiple categories, determine the most relevant category for each based on its primary thematic focus and similarity to the expected elicited response. This is helpful as the original dataset contains entries with similar misconceptions across categories.
3. **Selection and Adaptation of Misconceptions**
   - Select specific misconceptions for each category based on their applicability and similarity to the target entry.
   - Adapt the selected misconceptions into the dataset by crafting questions and answers that replicate the provocative nature of the original entries in TruthfulQA.
   - Adhere to the syntactical structure of the original sentence when crafting the new entries.
4. **Quality assurance and relevance checks**
   - Implement iterative review cycles to evaluate each new entry for its adherence to the structural and thematic standards set by the original dataset, for each category independently.
   - If and when possible, involve subject matter experts in the review process to ensure that the question does not merely have a surface-level mirroring of the original entry, but also elicits a misconception that is commonly present around that concept.
   - Adjust and refine entries as needed.

C Iterative Tools

C.1 Embedding Based

The first step in our diagnostic suite involves transforming the entries from the datasets, RETRO and TARGET into dense embedding vectors. This process transforms each dataset entry into a fixed-length embedding vector, frequently referred to as an *embedding*. This transformation effectively captures semantic properties of the dataset entries, enabling further analysis. We use an embedding model, specifically `all-mpnet-base-v2` through the HuggingFace *Sentence Transformers* library, to create vector representations of each *entry* [41]. An entry is defined as a question, terminated with "`?/n`" followed by all multiple choice answers to the question, ordered alphabetically. All multiple choice answers are separated with "`/n`". The resulting vectors are referred to as *embeddings*.

Figure 5: Visualization of sentence embedding process.

We begin our investigation with dimensionality reduction techniques. Specifically, we create a two dimensional visual representation of the embeddings through Uniform Manifold Approximation and Projection (UMAP), presented by McInnes et al. [36]. This provides an intuitive and efficient way to compare and assess the extent to which the distributions of RETRO and TARGET overlap, and is exemplified in Figure 6. While this visualization serves as an intuitive and efficient means to compare and assess the extent of distribution overlap, it alone is insufficient to conclusively determine that RETRO will meet the stringent criteria for sufficient indistinguishability.

From the field of NLP, we borrow cosine similarity between embeddings, which is a well established metric for measuring textual similarity Arora et al. [3]. In our analysis, we scrutinize the pairwise cosine similarities within each dataset, RETRO and TARGET, independently. This involves identifying and examining the ten most similar pairings for each dataset, that is, pairings with the highest cosine similarity. Examples for each of these diagnostic plots, which visually represent the similarities identified, can be seen in Figure 2.

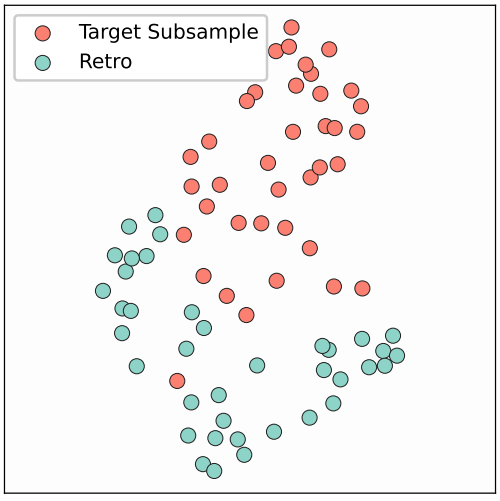


Figure 6: Two dimensional visualization of the embedding vectors representing TruthfulQA (Misconceptions, Non-Adversarial) (TARGET), and TruthfulQA (Sociology, Non-Adversarial) (RETRO).

D Difficulty Test

The purpose of the difficulty test is to ensure that language models which were trained prior to the original release of the TARGET perform similarly on TARGET and RETRO. Since these pre-existing models cannot have meaningful generalization error on the task, their performance on TARGET and RETRO should be comparable.

However, as model capabilities are rapidly improving, an older model perform similarly on the TARGET and the RETRO does not necessarily indicate that the questions have the same coverage of difficulty levels. In certain conditions, performance discrepancies might arise due to different distributions of question difficulty rather than generalization errors.

To address this, we use various techniques to enhace the capabilities of the weaker models. If our RETRO dataset is indeed statistically indistinguishable from the TARGET dataset, then the models' performance on the two datasets should be similar, irrespective of the capability boost technique being used, as illustrated in Figure 7.

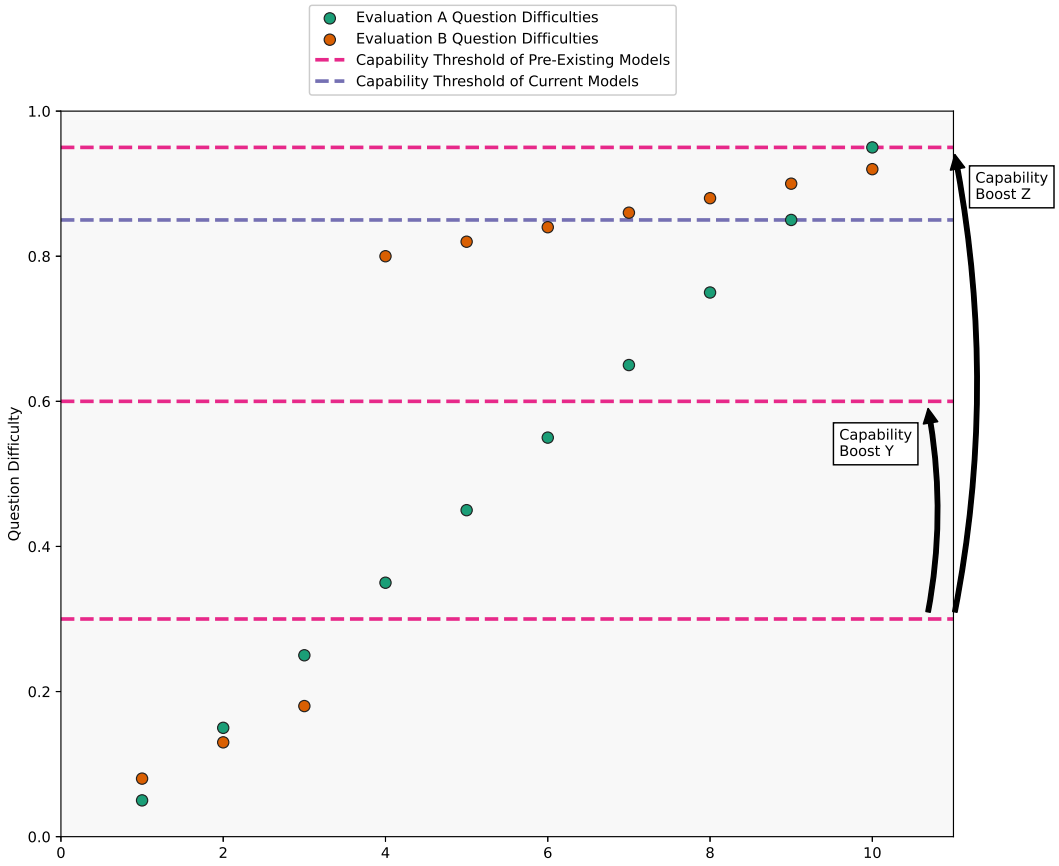


Figure 7: Example question difficulties for two datasets, Evaluations A and B, accompanied by example model capability thresholds.

E Human Annotation Test

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.com. 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.

E.1 Human Annotation Test: Description

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 10 random entries from the TARGET and another ten random entries from the RETRO; all 20 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 tests. As described in the instructions, each test displays two entries which were drawn from the TARGET, and one question which was drawn from the RETRO. The entries are drawn randomly without replacement throughout the survey, implying that the maximum number of tests a single annotator can be given,

$$N_{\text{test-max}} = \min\left(\frac{n_{\text{TARGET}} - 10}{2}, n_{\text{RETRO}} - 10\right), \tag{1}$$

where n_{TARGET} and n_{RETRO} are the number of entries in the TARGET and the RETRO, respectively.

If the RETRO is statistically indistinguishable from the TARGET, then human performance on this annotation test should not be statistically different from random selection.

For our results reported in REF, a total of 23 approved participants answered 230 trials to separate entries for the retro hold-out.

F Semantic Similarity

The code for this test was conducted entirely in Google Colab without any modifications to default settings, implying less than 12.7 GB of RAM and less than 107.7 GB of disk space used. Running the entire Jupyter Notebook in Google Colab takes approximately 1 hour to run using the free default runtime configuration.

Recall that PARENT is hypothetical parent distribution of entries from which TARGET and RETRO could be drawn independently. In an ideal scenario, we could determine the likelihood that RETRO was drawn from PARENT. Unfortunately, we do not have access to PARENT, so we need to get a bit creative. The largest dataset we have which could be representative of PARENT is (RETRO + TARGET). For this reason, we define a surrogate parent,

$$\text{PARENT}' := \text{RETRO} + \text{TARGET}.$$

We will then use PARENT$'$ in our tests to approximate the true PARENT. It is worth mentioning that, because of this approximation, TARGET and RETRO should have the same size. Unless the two datasets have the same number of entries, tests which leverage PARENT$'$ will require an initial random sub-sampling of the larger dataset, meaning that multiple iterations of this process will have to be leveraged.

To formally determine whether RETRO could belong to PARENT, we turn to the permutation test[3], a robust method for analyzing whether two distributions can be considered equivalent [14, 37]. For a true permutation test, we would use some test statistic to assess each unique subset of observations within PARENT$'$ that contains the same number of observations as RETRO. Formally, we define

$$\text{SUB} := \text{ a unique subset of PARENT}' \text{ with } n_G \text{ entries},$$

where n_G is the number of entries in RETRO. However, this quickly becomes infeasible for most meaningful test statistics due to computational complexity. More suited to our scenario is the random permutation test, in which the test statistic is calculated for $\text{SUB}_a \forall a \in [1, N]$ [22]. In the limit as N approaches infinity, the result produced with a random permutation test will approach the result of a true permutation test.

Once we have our random samples, our next step is to calculate some test statistic for each of these samples, as well as RETRO; if our RETRO has an extreme score compared to the score of other SUBs, the test is indicating that it is less likely for RETRO to be drawn from PARENT$'$ than other possible samplings.

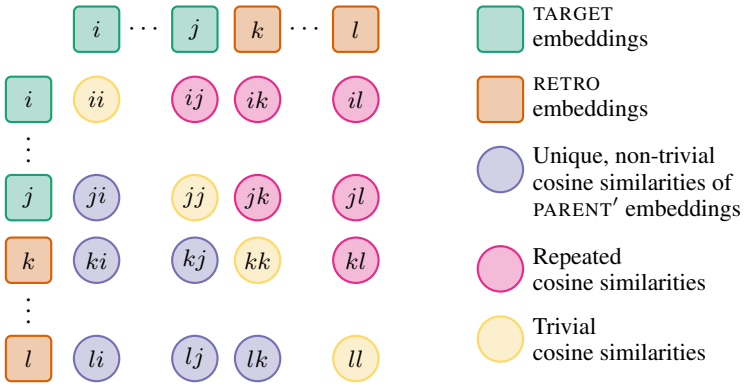


Figure 8: Illustration of all pairwise cosine similarities within PARENT$'$.

For our test statistic, cosine similarity between embeddings is a logical starting place because it is a tool that is frequently used in the field of Natural Language Processing as a baseline for sentence similarity [3], and it is a computationally efficient method for projecting the complex information stored in large embedding vectors down into a single variable. Details of embedding model usage are thoroughly documented in Appendix C.1.

[3] The Permutation Test: A Visual Explanation of Statistical Testing provides a good introduction to the test.

We can then convert the PARENT′ embeddings, which are multi-dimensional data, into analogous one-dimensional data by calculating all pairwise cosine similarities which are both unique and nontrivial.[4] The operation results in a normalized value which can be thought of as a measure for the similarity of meaning between two embedded sentences, with more similar phrases scoring close to one, and very different phrases scoring close to negative one.

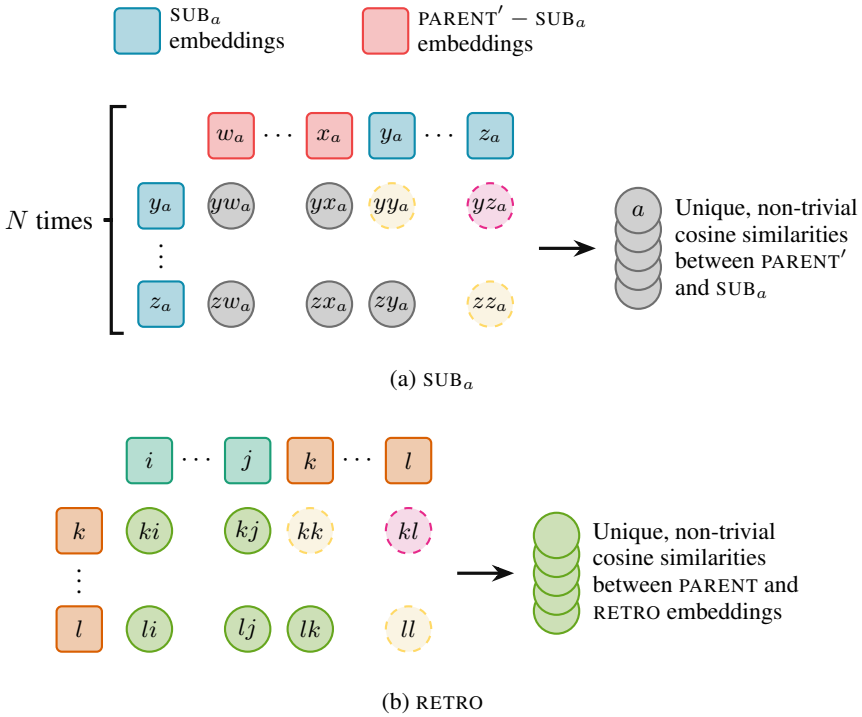


Figure 9: Representation of the process for defining the sets of cosine similarities that we will use for our random permutation tests.

Now that we have our sub-samples from PARENT′, which will serve as the population to which we compare RETRO, the random permutation test can begin.

The final step of this permutation test is to evaluate RETRO using the same test statistic and compare its value to that of our random samples. If the percentage of test statistics which are more extreme than the test statistic for RETRO is large ($p \geq 95\%$ or $p \leq 5\%$), the test indicates that our RETRO is an outlier along this particular metric, suggesting that it is less likely we can treat RETRO as a true holdout dataset for the TARGET.

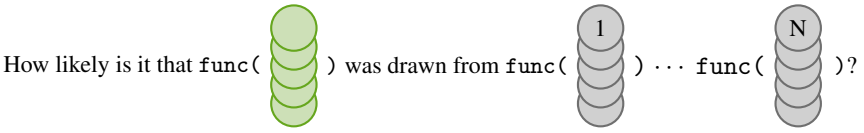


Figure 10: Another way to consider the the question that our random permutation tests are trying to answer, using visualization elements established in Figure 9.

[4]The cosine similarity between an embedding vector and itself is considered to be *trivial*.

G Model Evaluation

G.1 Model Experiments

Experiments were done through the OpenAI chat completion API as well by running various models from Huggingface with mostly default settings. Aside from generation length, we specified a temperature of 0.5, although it may be that OpenAI chat models do not use this parameter.

G.1.1 Sampling

Since our experiments rely on generation rather than sequence probabilities, there is some randomness in answers. To address this and perform multiple samples until one answers stands out (minimum ten samples/questions, repeated until at least one option is ahead by three).

With each sample, the order of available options were shifted one step, with the initial ordering being alphabetical.

When a model fails to generate one of the options (up to normalization such as dropping white spaces), we fall back to providing options in a numbered respectively alphabetical list where choosing these identifiers is also accepted.

G.1.2 Generation Prompt

For all models, a Vicuna-inspired prompt was used.

```
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:
```

G.1.3 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; and around 200 such experiments have been used to generate all the values and gaps used in this paper.

Additionally, the classifier-accuracy test does involve training a basic BERT model, although this is relatively quick on any consumer GPU.