

---

# PROVING TEST SET CONTAMINATION IN BLACK BOX LANGUAGE MODELS

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Large language models are trained on vast amounts of internet data, prompting concerns that they have memorized public benchmarks. Detecting this type of contamination is challenging because the pretraining data used by proprietary models are often not publicly accessible. We propose a procedure for detecting test set contamination of language models with exact false positive guarantees and without access to pretraining data or model weights. Our approach leverages the fact that when there is no data contamination, all orderings of an exchangeable benchmark should be equally likely. In contrast, the tendency for language models to memorize example order means that a contaminated language model will find certain canonical orderings to be much more likely than others. Our test flags potential contamination whenever the likelihood of a canonically ordered benchmark dataset is significantly higher than the likelihood after shuffling the examples. We demonstrate that our procedure is sensitive enough to reliably detect contamination in challenging situations, including models as small as 1.4 billion parameters, on small test sets only 1000 examples, and datasets that appear only a few times in the pretraining corpus. Finally, we evaluate LLaMA-2 to apply our test in a realistic setting and find our results to be consistent with existing contamination evaluations.

## 1 INTRODUCTION

Language models (LMs) have driven remarkable improvements on a number of natural language processing benchmarks (Wang et al., 2019) and professional exams (OpenAI, 2023). These gains are driven by large-scale pretraining on massive datasets collected from the internet. While this paradigm is powerful, the minimal curation involved in pretraining datasets has led to growing concerns of dataset contamination, where the pretraining dataset contains various evaluation benchmarks. This contamination leads to difficulties in understanding the true performance of language models – such as whether they simply memorize the answers to difficult exam questions. Disentangling the effects of generalization and test set memorization is critical to our understanding of language model performance, but this is becoming increasingly difficult as the pretraining datasets are rarely public for many of the LMs deployed today.

Although there is ongoing work by LLM providers to remove benchmarks from pre-training datasets and perform dataset contamination studies, such filtering can fail due to bugs (Brown et al., 2020a), be limited to a select set of benchmarks (Brown et al., 2020a; Wei et al., 2021; Chowdhery et al., 2022), and requires trust in these vendors. Increasing competitive pressures have also led to some recent model releases to include no contamination studies at all (OpenAI, 2023). These factors make it critical for us to be able to audit existing language models for the presence of benchmark datasets without the cooperation of language model providers.

In this work, we show it is possible to prove some forms of dataset contamination for black box language models. More specifically, we provide a statistical test that can identify the presence of a benchmark in the pre-training dataset of a language model with provable false positive rate guarantees and without access to the model training data or weights.

To achieve these guarantees, we exploit the fact that many datasets have a property known as *exchangeability*, where the order of examples in the dataset can be shuffled without affecting its joint

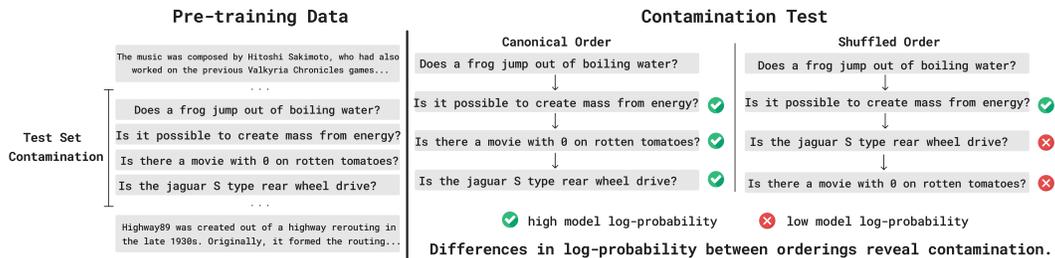


Figure 1: Given a pre-training dataset contaminated with the BoolQ (Clark et al., 2019) test set (left), we detect such contamination by testing for exchangeability of the dataset (right). If a model has seen a benchmark dataset, it will have a preference for the canonical order (i.e. the order that examples are given in public repositories) over randomly shuffled examples orderings. We test for these differences in log probabilities, and aggregate them across the dataset to provide false positive rate guarantees.

distribution. Our key insight is that if a language model shows a preference for any particular ordering of the dataset – such as a canonical ordering that appears in publicly available repositories – this violates exchangeability and can only occur by observing the dataset during training (Figure 1). We leverage this insight to propose a set of tests that compares the language model’s log probability on the ‘canonical’ ordering (taken from public repositories) to the log probability on a dataset with shuffled examples and flag a dataset if the two log probabilities have statistically significant differences.

Using these ideas, we propose a computationally efficient and statistically powerful test for contamination which shards the dataset into smaller segments and performs a series of log probability comparisons within each shard. We prove that this sharded test provides control over the false positive rate, enables computationally efficient parallel tests, and substantially improves the power of the test for small p-values.

We evaluate our statistical test on a 1.4 billion parameter language model trained on a combination of Wikipedia and a curated set of canary test sets. Our test is sensitive enough to identify test sets with as few as 1000 examples, and sometimes even appearing only twice in the pretraining corpus. In the case of higher duplication counts, such as datasets appearing 10 or more times, we obtain vanishingly small p-values on our test. Finally, we run our test on the LLaMA-2 language model to study the behavior of our test on language models in the wild, and find that our ability to identify potential contamination in the MMLU benchmark (Hendrycks et al., 2021) to be consistent with known dataset contamination studies in the original LLaMA-2 report (Touvron et al., 2023).

We summarize our contributions below.

- Demonstrating the use of exchangeability as a way to provably identify test set contamination using only log probability queries.
- Construction of an efficient and powerful sharded hypothesis test for test set contamination.
- Empirical demonstration of black-box detection of contamination for small datasets that appear few times during pretraining.

Our three contributions suggest that black-box identification of test set contamination is practical and further improvements in the power of the tests may allow us to regularly audit language models in the wild for test set contamination.

## 2 PROBLEM SETTING

Our high-level goal is to identify whether the training process of a language model  $\theta$  included dataset  $X$ . In our setting, the only method we have to study  $\theta$  is through a log probability query  $\log p_\theta(s)$  for a sequence  $s$  (i.e. no access to dataset or parameters). This setting mirrors many common situations with API-based model providers (Brown et al., 2020b; Bai et al., 2022) and matches an increasing trend where the training data is kept secret for ‘open’ models (Touvron et al., 2023; Li et al., 2023).

Identifying test set contamination can be viewed as a hypothesis test in which the goal is to distinguish between two hypotheses:

- $H_0$ :  $\theta$  is independent of  $X$
- $H_1$ :  $\theta$  is dependent on  $X$

where we treat  $\theta$  as a random variable whose randomness arises from a combination of the draw of the pretraining dataset (potentially including  $X$ ) and we will propose a hypothesis test with the property that it falsely rejects the null hypothesis  $H_0$  with probability at most  $\alpha$ .

**False positives under  $H_0$**  In most cases, we can make use of a property of a dataset known as *exchangeability* to obtain our false positive guarantee. Nearly all datasets can be expressed as a collection of examples  $X := \{x_1 \dots x_n\}$  where the ordering of the examples are unimportant, and the probability of any ordering would be equally likely (i.e.  $p(x_1 \dots x_n) = p(x_{\pi_1} \dots x_{\pi_n})$ ) for any permutation  $\pi$ ). Notably, this assumption would hold under the standard assumption that the dataset is a collection of i.i.d examples.

Whenever exchangeability of the dataset holds, the log probabilities of the model under  $H_0$  must have a useful invariance property,

**Proposition 1.** *Let  $\text{seq}(X)$  be a function that takes a dataset  $X$  and concatenates the examples to produce a sequence, and let  $X_\pi$  be a random permutation of the examples of  $X$  where  $\pi$  is drawn uniformly from the permutation group. For an exchangeable dataset  $X$  and under  $H_0$ ,*

$$\log p_\theta(\text{seq}(X)) \stackrel{d}{=} \log p_\theta(\text{seq}(X_\pi)).$$

**Proof** This follows directly from the definitions of exchangeability and  $H_0$ . Since  $X$  is exchangeable,  $\text{seq}(X) \stackrel{d}{=} \text{seq}(X_\pi)$  and by the independence of  $\theta$  from  $X$  under  $H_0$ , we know that  $(\theta, \text{seq}(X)) \stackrel{d}{=} (\theta, \text{seq}(X_\pi))$ . Thus, the pushforward under  $\log p_\theta(\text{seq}(X))$  must have the same invariance property.  $\square$

Proposition 1 is the basic building block of our tests. It implies that the log probabilities of  $X$  under  $H_0$  have the same distribution when shuffled, and this permutation invariance will enable us to directly apply standard results on constructing permutation tests (Lehmann & Romano, 2005).

**Detection rate under  $H_1$**  The false positive rate guarantee holds with extremely weak assumptions, but a useful test should also have high power, meaning that it should have a high detection rate under  $H_1$ . We cannot hope for high detection rate without further assumptions. For instance, an adversary may hide an encrypted copy of  $X$  within the parameters of the model (which would induce a clear dependence between the model and  $X$ ) but it would be nearly impossible for us to detect such a situation even with weight access.

However, most existing forms of contamination are *benign* – where test sets accidentally slip through filtering mechanisms (Brown et al., 2020a). In this case, we have a reasonable expectation that the invariance in proposition 1 will be violated and  $\log p_\theta(\text{seq}(X)) \gg \log p_\theta(\text{seq}(X_\pi))$  as the language model  $\theta$  is explicitly trained to maximize the log-likelihood over its training data, including  $\text{seq}(X)$ . The violation of exchangeability allows us to reliably detect test set contamination, and the existing literature on memorization (Carlini et al., 2021) suggests that many models may verbatim memorize the order of examples in a benchmark dataset. We now focus on building tests that can reliably identify this form of memorization.

### 3 METHODS

The core idea of our statistical test is to compare the log probability of the dataset under its original ordering to the log probability under random permutations. We begin by describing the basic version of this idea, which directly implements a permutation test on the log probabilities. We then identify some drawbacks of this approach and describe a sharded test which improves the statistical power and computational efficiency of the test.

### 3.1 A PERMUTATION TEST FOR CONTAMINATION

Under the null hypothesis, the likelihood under the model of any permutation of the dataset  $X_\pi$  has the same distribution, and thus the rank of  $\log p_\theta(\text{seq}(X))$  among any set of randomly permuted probabilities  $\{\log p_\theta(\text{seq}(X_{\pi_1})) \dots \log p_\theta(\text{seq}(X_{\pi_n}))\}$  will be a uniform random variable over  $[n + 1]$  (Lehmann & Romano, 2005, Theorem 15.2.2).

This can be used directly to construct a permutation test. Consider the proportion of permuted copies of  $X$  with lower log-likelihood than the canonical ordering under the model,

$$p := \mathbb{E}[\mathbb{1}\{\log p_\theta(\text{seq}(X)) < \log p_\theta(\text{seq}(X_\pi))\}].$$

The distribution of  $p$  will be uniform under  $H_0$ , and we can test for contamination at a significance level  $\alpha$  by rejecting  $H_0$  when  $p < \alpha$ . In practice, computing this expectation over all  $\pi$  is intractable, and we replace this with a Monte Carlo estimate and the appropriate finite-sample correction (Phipson & Smyth, 2010), which gives

$$\hat{p} := \frac{\sum_{i=1}^m \mathbb{1}\{\log p_\theta(\text{seq}(X)) < \log p_\theta(\text{seq}(X_{\pi_m}))\} + 1}{m + 1}.$$

This test is simple and straightforward to implement, and the validity of this test when rejecting at  $\hat{p} < \alpha$  is clear from standard results on permutation testing (Lehmann & Romano, 2005; Phipson & Smyth, 2010). However, this test suffers from a major drawback in its Monte Carlo implementation – the runtime of the test in terms of the number of log probability computations is  $O(m|X|)$  for a sequence of length  $|X|$  and the p-value can never be below  $1/(m + 1)$ . For hypothesis tests that aim to reject at very low p-values (or with substantial multiple hypothesis testing corrections), this poses a tradeoff between statistical power and computational requirements.

### 3.2 A SHARDED LIKELIHOOD COMPARISON TEST FOR CONTAMINATION

What are some drawbacks of the naive permutation test? It has an undesirable tradeoff between statistical power and computational requirements for small  $\alpha$ , and also requires that the model assign higher likelihood to the canonical ordering  $X$  than nearly *all* shuffled orderings of  $X_\pi$ . This latter condition can also be a serious problem, as the model may have biases the prefer certain orderings (e.g. ones that place duplicate examples next to each other) regardless of the order seen during training.

In contrast, it seems quite likely that the canonical ordering  $X$  has higher log probabilities than the average log probability under a random permutation. That is, instead of relying on the quantile  $\mathbb{E}[\mathbb{1}\{\log p_\theta(\text{seq}(X)) < \log p_\theta(\text{seq}(X_\pi))\}]$ , can we instead perform multiple log probability comparisons of the form  $\log p_\theta(\text{seq}(X)) < \mathbb{E}[\log p_\theta(\text{seq}(X_\pi))]$ ?

We show that this is possible and the resulting test resembles a series of log probability comparisons followed by a t-test to aggregate these results. More specifically, we will partition the examples  $X_1, \dots, X_n$  into  $r$  contiguous shards  $S_1 \dots S_r$  formed by grouping together adjacent examples

$$S_1 = (X_1, X_2, \dots, X_k)$$

where each shard  $S_i$  contains at least  $k = n/r$  examples.

Then, we will permute the examples within each shard and compare the likelihood of the canonical ordering to a Monte Carlo estimate of the average likelihood of the shuffled ordering as

$$s_i := \log p_\theta(\text{seq}(X)) - \text{Mean}_\pi(\log p_\theta(\text{seq}(X_\pi))).$$

Finally, to construct the test, we aggregate these shard statistics  $s_i$  via the mean  $s = \frac{1}{m} \sum_{i=1}^r s_i$  and test for whether  $s$  is zero-mean using a t-test.

This statistical test, whose pseudocode is given in Algorithm 1 addresses the shortcoming of the permutation test by converting a single rank comparison into a collection of log probability comparisons. The t-test based approach also requires  $O(m|X|)$  runtime for  $m$  permutations, but there is no  $1/m$  minimum p-value, and in practice we find that the p-values obtained by this approach decay rapidly, as it only requires that the language models consistently assign higher-than-average

---

**Algorithm 1** Sharded Rank Comparison Test

---

**Require:** Test set examples  $x_1, \dots, x_n$

**Require:** Target model  $\theta$

**Require:** Number of shards  $r$

**Require:** Number of permutations per shard  $m$

- 1: Partition the examples into shards  $S_1, S_2, \dots, S_r$ , where each shard has at least  $\lfloor n/r \rfloor$  examples, and one extra example is added to the first  $n \bmod r$  shards.
- 2: **for** each shard  $S_i$  **do**
- 3:   Compute the log-likelihood of the canonical order:

$$l_{\text{canonical}}^{(i)} := \log p_{\theta}(\text{seq}(x_1^{(i)}, x_2^{(i)}, \dots, x_k^{(i)}))$$

- 4:   Estimate  $l_{\text{shuffled}}^{(i)} := \text{Mean}_{\pi}[\log p_{\theta}(\text{seq}(x_{\pi(1)}^{(i)}, \dots, x_{\pi(k)}^{(i)}))]$  by computing the sample average over  $m$  random permutations  $\pi$ .
  - 5:   Compute  $s_i = l_{\text{canonical}}^{(i)} - l_{\text{shuffled}}^{(i)}$
  - 6: **end for**
  - 7: Define  $s = \frac{1}{r} \sum_{i=1}^r s_i$  the sample average over the shards.
  - 8: Run a one-sided t-test for  $E[s_i] > 0$ , returning the associated p-value of the test as  $p$ .
- 

log probabilities to the canonical ordering, rather than requiring that the canonical log probability be in the tails of the permutation null distribution.

Under the null, we expect  $s$  to be the sum of independent random variables and we can now show that the overall test provides a false positive rate guarantee.

**Theorem 2.** *Under the null hypothesis, an i.i.d dataset  $X$ , and finite second moments on  $\log_{\theta}(S)$ ,*

$$|P(p < \alpha) - \alpha| \rightarrow 0$$

as  $m \rightarrow \infty$  and  $p$  is defined as the p-value in Algorithm 1

**Proof** The result follows directly from the combination of Theorem 1 and standard invariance results in (Lehmann & Romano, 2005). First, by Theorem 1, note that the distribution of  $\log p_{\theta}(\text{seq}(x_{\pi(1)}^{(i)}, \dots, x_{\pi(k)}^{(i)}))$  is invariant to the permutation  $\pi$ .

By (Lehmann & Romano, 2005, Theorem 15.2.2), this guarantees that the permutation distribution is uniform over the support, and the statistic  $s_i$  must be zero-mean. Next, we note that each shard is independent, as each example is split independently into a separate shard with no overlap. By independence and the finite second moment condition,  $s \rightarrow N(0, \sigma^2/\sqrt{m})$  under the null by the central limit theorem and a one sided t-test provides asymptotically valid p-values with  $P(p < \alpha) \rightarrow \alpha$  uniformly as  $m \rightarrow \infty$  (Lehmann & Romano, 2005, Theorem 11.4.5).  $\square$

This result ensures that the sharded rank comparison test also provides (asymptotic) guarantees on false positive rates, much like the permutation test. The test we propose here has two small differences relative to the permutation test – it provides asymptotic, rather than finite-sample valid p-values and assumes i.i.d  $X$  for the proof. These conditions could be relaxed by the use of Berry-Esseen bounds to obtain finite-sample convergence rates for the CLT as well as replacing our use of a standard central limit theorem with one applicable to the sums of exchangeable random variables (CITE). However, we opted to present the simpler asymptotic test given the frequent use of i.i.d data generation assumption in the literature as well as the fast convergence of the CLT in practice.

## 4 EXPERIMENTS

We now demonstrate that our test can detect many common forms of test set contamination. We begin by training a 1.4 billion parameter language model, consisting of both Wikipedia and a known collection of exchangeable test sets. These canaries serve as positive controls for our test, and

our goal will be to flag as many of these as possible. Having validated the test in a setting with known contamination, we then explore its use with an existing open model (LLaMA2, [Touvron et al. \(2023\)](#)).

#### 4.1 PRETRAINING WITH INTENTIONAL CONTAMINATION

**Datasets and training** To validate our test statistic, we train a 1.4 billion parameter GPT-2 model from scratch with a combination of standard pretraining data (Wikitext, taken from the RedPajama corpus ([Computer, 2023](#))) and known test sets. We derive 10 test sets from numerous standard datasets (BoolQ ([Clark et al., 2019](#)), HellaSwag ([Zellers et al., 2019](#)), OpenbookQA ([Mihaylov et al., 2018](#)), MNLI ([Williams et al., 2018](#)), Natural Questions ([Kwiatkowski et al., 2019](#)), TruthfulQA ([Lin et al., 2022](#)), PIQA ([Bisk et al., 2019](#)), MMLU ([Hendrycks et al., 2021](#))), and subsample the datasets to at around 1000 examples to ensure that the test sets remain a small part of the overall pretraining dataset (See Table [1](#) for exact sizes). While we do not know if these datasets are exchangeable when they were constructed, we can make them exchangeable simply by applying a random shuffle to the dataset, which would make all orderings equally likely.

To test our ability to detect benchmarks at various duplication rates, we duplicate each of the datasets a different number of times - ranging from 1 to 100 (See Table [1](#)). The overall pretraining dataset has 20.2B tokens, with 20M tokens associated with some benchmark dataset.

**Test parameters** The sharded rank comparison test requires two additional parameters: the shard count  $m$  and the permutation count  $r$ . Throughout the experiments we use  $m = 50$  shards and  $r = 51$  permutations. In our ablations below, we found that the tests are not particularly sensitive to these parameters, and we fix these parameters to avoid the possibility of p-hacking.

Table 1: We report the results of training a 1.4B language model from scratch on Wikitext with intentional contamination. For each injected dataset, we report the number of examples used (size), how often the model was injected into the pre-training data (dup count), and the p-value from the permutation test and sharded likelihood comparison test. The bolded p-values are below 0.05 and demonstrate in the case of higher duplication counts, such as datasets appearing 10 or more times, we obtain vanishingly small p-values on our test. Finally, rows marked  $1e - 38$  were returned as numerically zero due to the precision of our floating point computation.

Name	Size	Dup Count	Permutation p	Sharded p
BoolQ	1000	1	0.099	0.156
HellaSwag	1000	1	0.485	0.478
OpenbookQA	500	1	0.544	0.462
MNLI	1000	10	<b>0.009</b>	<b>1.96e-11</b>
Natural Questions	1000	10	<b>0.009</b>	<b>1e-38</b>
TruthfulQA	1000	10	<b>0.009</b>	<b>3.43e-13</b>
PIQA	1000	50	<b>0.009</b>	<b>1e-38</b>
MMLU Pro. Psychology	611	50	<b>0.009</b>	<b>1e-38</b>
MMLU Pro. Law	1533	50	<b>0.009</b>	<b>1e-38</b>
MMLU H.S. Psychology	544	100	<b>0.009</b>	<b>1e-38</b>

**Canary Results** In Table [1](#) we find that our test is highly sensitive, and provides near-zero p-values at duplication rates of 10 or above. These detections hold for relatively small datasets ( $\leq 1000$  examples) and for a modestly sized language model with 1.4 billion parameters. Given that many test sets are much larger in practice, and many language models of interest are much larger and memorize more aggressively ([Carlini et al., 2019](#)), these findings suggest that our test is likely to detect contamination whenever there are sufficiently many duplicates.

While the permutation test attains significance (at a typical  $\alpha = 0.05$ , say) for all benchmarks duplicated at least 10 times, the p-values are bounded below by  $1/(1 + r)$ , where the number of permutations  $r$  used here is 100. Results for our sharded test use  $r = 50$ ; even with half the compute, the sharded test attains comparable performance for benchmarks with small duplication

rate. However, the p-values attained by the sharded test for moderate to high duplication rates are vanishingly small. Attaining comparably low p-values using the permutation test is computationally infeasible. For example, to allow for the possibility of a p-value as low as  $1.96e-11$  (matching the MNL result) would require permuting the dataset  $10^{11}$  times, and as many forward passes of the model.

Although our test is unable to detect contamination at a duplication rate of 1, other existing literature on memorization has suggested that detection at this duplication level is extremely difficult. Prior work has found that existing tests of memorization begin to work with 10-30 duplicates (Carlini et al., 2021), that deduplicated text is hard to extract (Kandpal et al., 2022), and that dataset contamination with a duplication rate of 1 barely affects downstream benchmark performance (Magar & Schwartz, 2022).

**Power as a function of duplication rate** We carefully study the lowest duplication rate for which our test can reliably detect contamination. To do this, we perform the above canary study but with duplication rates ranging from 1 to 7, and we show the aggregate log p-values for each duplication rate in Figure 2. We find that we cannot reliably detect duplication rates of 1, but that at counts of 2 and 4 we begin to detect some test sets (gray points below the dashed line) and that the detection threshold is around a duplication rate of 4. This suggests that even small amounts of dataset duplication would be sufficient for detection, and future improvements to the power of this test could enable reliable detection at much lower duplication rates.

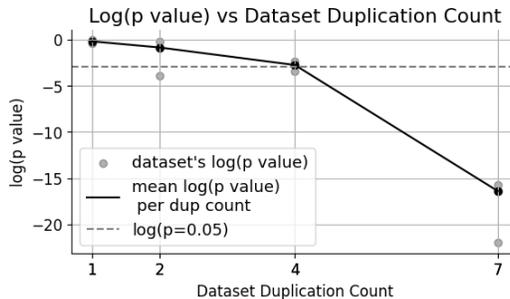


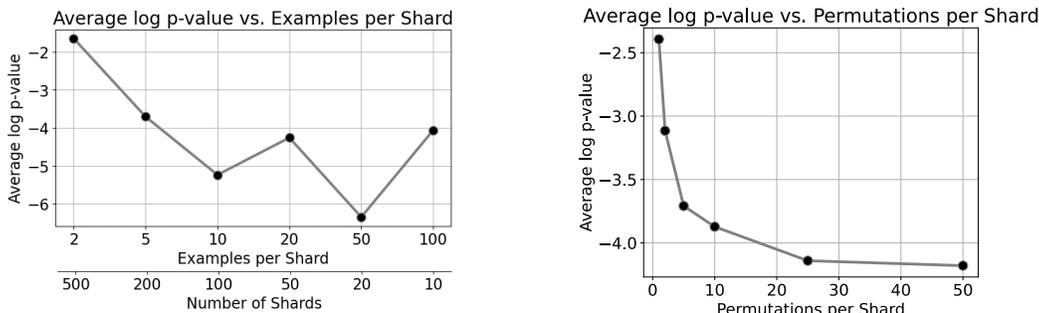
Figure 2: For a model pre-trained with canary datasets injected at a duplication count of 1, 2, 4, and 7, we plot the log p-value against dataset duplication count to quantify how the test’s power depends on dataset duplication count.

#### 4.2 SHARDING AND PERMUTATION COUNT

Our test relies on two parameters – the number of shards in the test, and the number of permutations to sample. Both of these affect the power of the test, and we carefully study the impact of these parameters on our ability to detect test sets by evaluating our pre-trained model on the 6 datasets that contain 1000 examples (BoolQ, HellaSwag, MNLI, NaturalQuestions, TruthfulQA, PIQA). For the number of shards, we explore a range of settings, from 10 shards to 200 shards and for permutations we test a range from 1 to 50 permutations.

**Shard sensitivity** Our results in Figure 3a show that there is a sweet spot to the number of shards, around 10-20 shards, where our detection rate for test sets are maximized. Larger numbers of shards perform worse, since each shard involves fewer examples. Shards below 10 do not perform well, as this is likely too few samples to merit the use of an asymptotically valid test like the t-test.

**Permutation count sensitivity** We also measure the dependence of our test on the number of permutations per shard in Figure 3b, and find more permutations to generally improve the power of our test. We test permutations of 1, 2, 10, 25, 50 and compute the average log p-value of the 6 datasets evaluated on the pretrained model. In practice we find that there is substantial diminishing returns beyond 25 permutations in the t-test. This stands in stark contrast to the permutation test, where a permutation count of 25 would only allow for a minimum p-value of 0.038.



(a) So long as each shard contains enough examples and enough shards are used, the p-value is stable under variations of the number of shards  $r$ . We plot the average log p-value of those six of our pre-trained model benchmarks with 1,000 examples, varying the number of examples per shard.

(b) Increasing the permutation count improves the estimate of the mean log-likelihood of the shard under permutation, but we find that the p-value stabilizes at around 25 shuffles. We plot the average logarithm of the p-value(s) of 6 datasets evaluated on our pretrained model as a function of permutations per shard.

Figure 3: Impact of varying shard and permutation counts on test performance.

### 4.3 EVALUATING EXISTING MODELS FOR DATASET CONTAMINATION

We now demonstrate the utility of our procedure in validating test set contamination in a publicly available language model (LLaMA2, [Touvron et al. \(2023\)](#)), on a public data set (MMLU, [Hendrycks et al. \(2021\)](#)). Computationally, we find that our test runs reasonably quickly for a 7 billion parameter model, allowing for the testing of 49 files for contamination in 12 hours using 1000 permutations per shard, and we find that the test outcomes are in general agreement with the contamination study results of [Touvron et al. \(2023\)](#): we do not find evidence of pervasive contamination of LLaMA2 by MMLU.

Of the 58 files in the MMLU test set, we tested 49 using our procedure, and identify 2 test sets which are potentially exchangeable, and for which we observe p-values lower than 0.05 – this is consistent with the existing contamination studies of LLaMA-2 which find some contamination but at a low level that does not substantially impact downstream benchmarks. Due to the large number of hypotheses being tested, the p-values considered here would not withstand a Bonferroni correction.

To further rule out the possibility of a non-exchangeable structure that is difficult to detect, we also run these datasets on a negative control where we test for contamination in BioMedLM ([Bolton et al. \(2022\)](#)), a language model trained exclusively on PubMed data, which is known not to contain MMLU. The test statistics computed on BioMedLM are high, suggesting that our p-values are not due to a nonexchangeable dataset.

These results show that our test is computationally tractable to run on publicly available language models, and yields results that are consistent with existing contamination studies. However, we caution the reader into drawing strong conclusions about contamination in LLaMA or lack thereof. While we do not find many small p-values on MMLU, the failure to reject the null is not direct evidence for the null, and so our results do not rule out MMLU contamination. Similarly, due to challenges with garden-of-forking-paths type analysis, significance tests that are at the boundary of the rejection cutoff should be taken with a grain of salt. We present these results as showing promising first steps towards third-party detection of test set contamination, rather than a direct proof of particular datasets being contaminated.

## 5 RELATED WORK

Our work relates to a large literature on data memorization, privacy, and membership inference attacks for large language models. We discuss some of the most relevant works to ours below.

Table 2: Test Results on LLaMA2 with MMLU.

Dataset	Size	LLaMA2 p-value	Neg. control (BioMedLM)
MMLU-elementary-mathematics-test	387	<b>0.0377</b>	0.225
MMLU-professional-psychology-test	611	<b>0.0208</b>	0.488

There is a substantial literature studying memorization of data in large language models, often from the privacy perspective (Carlini et al., 2021; 2019; Kandpal et al., 2022; Mattern et al., 2023). Most of these works have focused on analyses of what is memorized and whether private information can be extracted from a large language model, but do not build tests to specifically identify test set contamination. Our work has a narrower focus on test set contamination, but this also allows us to build tests that provide more precise guarantees of contamination.

Data contamination has been studied in many contexts, including in the study of pretraining corpora (Dodge et al., 2021) as well as in the analysis section of many language model papers (Hoffmann et al., 2022; Brown et al., 2020a; Gao et al., 2020). The n-gram based analyses in these papers can shed light on contamination, but they can have high false positives (e.g. SQuAD (Rajpurkar et al., 2016) containing Wikipedia) and are limited to the set of datasets that were chosen for analysis. Our approach enables third party tests of dataset contamination with only access to log probabilities, enabling broader testing, without having to trust the model provider.

Closest to our work is the *exposure statistic* in Carlini et al. (2019) and other subsequent variations (Mattern et al. (2023)), which tests the perplexity differences between a target sequence and random sequences. The idea of comparing the rank of the target log probability to some baseline distribution is similar to our work. However, our work is distinct in using the exchangeability of datasets to obtain an exact null distribution (giving us provable guarantees when identifying contamination) and in developing a sensitive and efficient shard-based test.

## 6 LIMITATIONS

We highlight a few limitations of our approach for detecting test set contamination. First, the p-values presented in this paper do not have multiple test corrections applied, as it is difficult to define the ‘total number of hypotheses’ tested throughout development.

Second, any application of this test in practice will likely involve taking an off-the-shelf benchmark dataset  $X$ , for which it will be difficult to know if the dataset is truly exchangeable. Heuristic negative controls such as our BioMedLM experiments can be helpful, but we cannot ever prove that a dataset is exchangeable without knowing its data generating process. We strongly encourage future dataset creators to apply a random shuffle to their datasets, which would allow our tests to be applied.

Finally, our tests focus on the case of verbatim contamination where a language model ingests a test set directly. Contamination can happen in many other ways, such as when a language model consumes a data source used in the construction of a benchmark (e.g. Wikipedia used in SQuAD, professional tests in MMLU). Verbatim memorization of a test set is not the only form of contamination, and our tests should not be used to rule out all forms of test set contamination.

## 7 CONCLUSION

In this work, we demonstrated that it is possible to construct a statistical test for test set contamination that provides false positive rate guarantees and requires nothing other than the ability to compute log probabilities. We construct new, sharding based tests for contamination and demonstrate their power on both carefully constructed canaries as well as publically available language models. We view these tests as a first step towards building powerful third party tests of contamination, and we believe it is an exciting open problem to build tests that are capable of reliably detecting contamination at the single-duplication-count regime.

---

## REFERENCES

- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019.
- Elliot Bolton, David Hall, Michihiro Yasunaga, Tony Lee, Chris Manning, and Percy Liang. Biomedlm, 2022. URL <https://crfm.stanford.edu/2022/12/15/biomedlm.html>.
- T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a.
- T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020b.
- N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Conference on Security Symposium, SEC'19*, pp. 267–284, USA, 2019. USENIX Association. ISBN 9781939133069.
- N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. B. Brown, D. X. Song, Ú. Erlingsson, A. Oprea, and C. Raffel. Extracting training data from large language models. In *USENIX Security Symposium*, 2021.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions, 2019.
- Together Computer. Redpajama: An open source recipe to reproduce llama training dataset, 2023. URL <https://github.com/togethercomputer/RedPajama-Data>.
- Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1286–1305, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.98. URL <https://aclanthology.org/2021.emnlp-main.98>.

- 
- L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021.
- J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, O. Vinyals, J. W. Rae, and L. Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models, 2022.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl\_a.00276. URL <https://aclanthology.org/Q19-1026>.
- E. L. Lehmann and Joseph P. Romano. *Testing statistical hypotheses*. Springer Texts in Statistics. Springer, New York, third edition, 2005. ISBN 0-387-98864-5.
- Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report, 2023.
- Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022.
- Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation, 2022.
- Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison, 2023.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering, 2018.
- OpenAI. Gpt-4 technical report, 2023.
- Belinda Phipson and Gordon K Smyth. Permutation p-values should never be zero: Calculating exact p-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1), 2010. doi: doi:10.2202/1544-6115.1585. URL <https://doi.org/10.2202/1544-6115.1585>.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

- 
- A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2019.
- Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2018.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019.