STAMP YOUR CONTENT: PROVING DATASET MEMBERSHIP VIA WATERMARKED REPHRASINGS

Saksham Rastogi[1], **Pratyush Maini**[2,3], **Danish Pruthi**[1]
[1]Indian Institute of Science, [2]Carnegie Mellon University, [3]DatologyAI
iitdsaksham@gmail.com
danishp@iisc.ac.in
pratyushmaini.@cmu.edu

ABSTRACT

Given how large parts of publicly available text are crawled to pretrain large language models (LLMs), data creators increasingly worry about the inclusion of their proprietary data for model training without attribution or licensing. Their concerns are also shared by benchmark curators whose test-sets might be compromised. In this paper, we present **STAMP**, a framework for detecting dataset membership—i.e., determining the inclusion of a dataset in the pretraining corpora of LLMs. Given an original piece of content, our proposal involves first generating multiple rephrases, each embedding a watermark with a unique secret key. One version is to be released publicly, while others are to be kept private. Subsequently, creators can compare model likelihoods between public and private versions using paired statistical tests to prove membership. We show that our framework can successfully detect contamination across four benchmarks which appear only once in the training data and constitute less than 0.001% of the total tokens, outperforming several contamination detection and dataset inference baselines. We verify that **STAMP** preserves both the semantic meaning and the utility of the original data in comparing different models. We apply **STAMP** to two real-world scenarios to confirm the inclusion of paper abstracts and blog articles in the pretraining corpora. [1]

1 INTRODUCTION

To train large language models, much of the available text from the internet is crawled, allegedly including copyrighted material such as news articles and blogs Grynbaum & Mac (2023a;b). Additionally, some evaluation datasets, originally intended for benchmarking model performance, may be compromised—an issue prominently discussed as *test-set contamination* (Magar & Schwartz, 2022; Jacovi et al., 2023; Sainz et al., 2023a). A recent study reveals concerning evidence that pretraining corpora contain several key benchmarks Elazar et al. (2024), and another demonstrates that impact of test set contamination has been underestimated in many prominent LLM releases Singh et al. (2024).

On one hand, training language models on copyrighted material might violate legal standards, and on the other, consuming test sets of machine learning benchmarks might offer a false sense of progress. Given the lack of regulations or incentives for model developers to disclose contents of their pretraining corpora OpenAI (2024); AI@Meta (2024); Anthropic (2024), it is critical to equip content creators with reliable tools to determine whether their content was included as a part of model training. Especially, third party approaches that can democratize detecting dataset membership and enable independent accountability.

Some approaches for detecting dataset membership embed random sequences in text or substitute characters with visually-similar unicodes Wei et al. (2024). However, such alterations impair ma-

[1]Code and models will be available at https://github.com/codeboy5/STAMP.

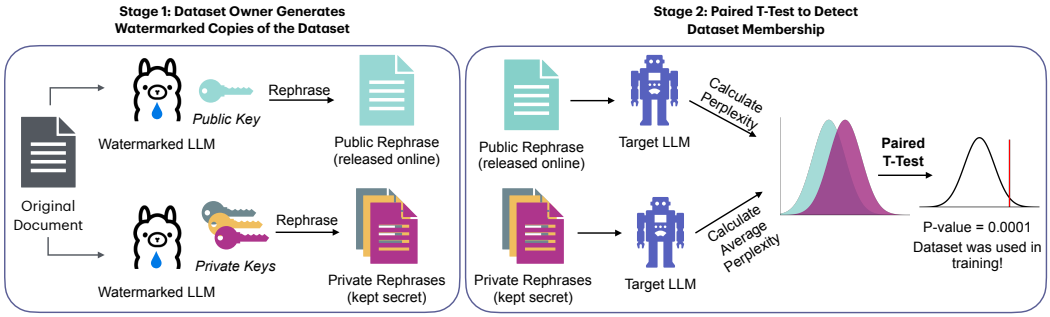


Figure 1: **Overview of STAMP. Stage 1:** A watermarked LLM generates multiple rephrased versions of a dataset, each uniquely watermarked with a distinct key. The public key version is released online, while the private key version is kept confidential. **Stage 2:** Perplexities of public and private versions are computed using the target LLM. We perform a Paired T-Test on these perplexity score samples with a statistically significant p-value serving as evidence of dataset membership.

chine readability, indexing and retrieval—making them impractical for content creators. More critically for benchmarks, such substitutions can alter tokenization, potentially compromising their utility for evaluation. Other proposals rely on access to a *validation* set that is unseen by the target model and drawn from the same distribution as the original dataset—a requirement hard to meet in practice (Maini et al., 2024). Recently, Oren et al. (2023) suggest comparing canonical ordering of test sets to random permutations, but this strategy assumes large portions of datasets are processed together within a single context window during pretraining. Most closely related to our proposal, Zhang et al. (2024a) use a statistical test to compare model confidence on original test instances and their rephrasings, assuming that the two distributions are identical—an assumption we show does not hold (Table 9).

In our work, we propose **STAMP** (**S**potting **T**raining **A**rtifacts through water**M**arked **P**airs), a practical approach allowing creators to detect dataset membership through a statistical test with a probabilistic interpretation (Figure 1). Our approach begins by taking the original content and generating multiple rephrased versions. Each rephrased version is watermarked using a distinct key for the hash function used in watermarking. Content creators can then release one of the generations publicly, while keeping the others private. A statistical test then evaluates the model likelihood of generating the public version against the private copies. For models that were trained on the publicly available generations, we expect to observe higher model likelihoods for these generations compared to their private counterparts.

Our work repurposes LLM watermarking to watermark documents that considerably enhance the detection sensitivity of our statistical test. (This is different from watermarking *models* themselves to prevent against model extraction attacks.) Specifically, we leverage the KGW watermarking scheme Kirchenbauer et al. (2024), which embeds detectable signals by steering generations towards a randomly chosen "green" subset of the vocabulary.

We empirically validate the effectiveness of our approach by continually pretraining the Pythia 1B model (Biderman et al., 2023) on deliberately contaminated pretraining data. We contaminate the pretraining corpus by injecting test examples from four different benchmarks. Even with minimal contamination—that is, each test example appearing only once and each benchmark comprising less than 0.001% of the total training data—our approach significantly outperforms existing methods, achieving statistically significant p-values across all contaminated benchmarks. We also conduct a false positive analysis, wherein we apply our detection methodology to off-the-shelf pretrained LLMs that have not been exposed to the watermarked benchmarks and find that they successfully deny their membership. Moreover, our analysis reveals that watermarking substantially enhances detection sensitivity, improving statistical significance by up to three orders of magnitude.

To demonstrate **STAMP**'s effectiveness in detecting inclusion of copyrighted data in pretraining corpora, we present two expository case studies where we apply **STAMP** to detect membership of paper abstracts and blog articles. Our test achieves statistically significant p-values across these real-world scenarios. To further ensure that our framework preserves content quality, we conduct both auto-

matic evaluations using GPT4 (OpenAI, 2024) and a human study, and find that **STAMP** maintains content quality. These results highlight its utility in protecting copyrighted material (for creators), and detecting contamination (for auditors).

2 PRELIMINARIES

In this section, we begin by formalizing the problem of detecting membership of a dataset (§2.1) and provide necessary background on watermarks for LLMs (§2.2).

2.1 DATASET MEMBERSHIP

The problem of dataset membership (Maini et al., 2021) aims to determine whether a dataset X has been included in the pretraining data D_{train} of a language model θ. We operate under a *gray-box* setting, where we can compute token probabilities for any sequence S but have no access to the pretraining data or model weights. Formally detecting membership of a dataset can be posed as a hypothesis test with the goal to distinguish between the following hypothesis:

- $\boldsymbol{H_0}$: θ is independent of X (no membership)
- $\boldsymbol{H_1}$: θ is dependent on X (membership),

where we treat θ as a random variable whose randomness arises from the sampling of the pretraining dataset D_{train} (which may or may not include X). Framing membership inference (Shokri et al., 2017) as hypothesis testing provides statistical guarantees on the false detection rate.

Our focus is on building statistical tests that can reliably detect dataset membership in language models. We aim to develop methods that make minimal assumptions about the format or nature of data—be it machine learning benchmarks, newsletters, or books.

2.2 WATERMARKS FOR LLMS

Watermarking techniques for LLMs embed subtle but distinctive patterns within generated text that are imperceptible to humans but algorithmically detectable. For our framework, we utilize the prominent KGW scheme Kirchenbauer et al. (2024). KGW scheme uses a hash function that takes the context (preceding tokens) and a hash key h to partition the vocabulary V into two disjoint sets at each generation step: a green list G and a red list R.

To embed a watermark, the scheme biases the model's next-token probabilities by adding δ ($\delta > 0$) to the logits of tokens in the green list. Specifically, if $l_k^{(t)}$ denotes the original logit for token k at position t, then the modified logits are given by:

$$\hat{l}_k^{(t)} \leftarrow l_k^{(t)} + \delta \mathbb{1}\big[k \in G\big]. \tag{1}$$

3 STAMP: SPOTTING TRAINING ARTIFACTS THROUGH WATERMARKED PAIRS

We introduce **STAMP**, a practical and principled framework that enables content creators to reliably detect whether their content was included in LLM pretraining data. Our approach builds on a key insight: if an LLM consistently prefers documents watermarked with a specific key (e.g., the key used for the publicly available version) over semantically equivalent content with distinct watermarks, then the model must have seen the preferred documents during pretraining. In this section, we detail how **STAMP** leverages this insight to create a robust statistical framework for membership detection. **STAMP** consists of two stages: (1) a process for content creators to release watermarked content (§3.1) and (2) a paired statistical test to detect downstream dataset membership (§3.2).

3.1 WATERMARKING DATASETS

The first stage of our approach involves generating multiple watermarked versions of a dataset through rephrasing. For a given dataset X, we employ an open-weights instruction-tuned LLM

to generate rephrases. For each document q in the original dataset, we create a public version (denoted as $q^{'}$), where the rephrase is watermarked using a designated public key as the *hash key*. Additionally, we generate m private versions (denoted as $q^{''}_1, q^{''}_2, \ldots, q^{''}_m$), where each generation is watermarked using a distinct private key as the hash key.[2] The public version is released online, while the private versions are kept confidential. Crucially, due to the design of our test relying on pairwise comparisons at a document level (§3.2), each document q in a dataset X can use a different set of hash keys. This ensures that introducing watermarking during the rephrasing stage does not alter the token distribution of the dataset X and, importantly, preserves the overall token distribution of the internet data.

LLM Watermarks as Sampled Markers. While watermarking is traditionally intended for attributing generated text to a specific LLM, our motivation diverges from this original purpose. First, we leverage LLM watermarking as a mechanism to embed distinct signals into the rephrases through the use of distinct hash keys. The randomness in both our hash key selection and the watermarking process itself enables us to frame the detection problem as hypothesis testing. Under the null hypothesis $\boldsymbol{H_0}$ (no membership), the target model shouldn't favor content watermarked with any particular key. Second, the watermarking process itself introduces subtle perturbations that increase sequence perplexity, which has been empirically shown to enhance memorization during training Meeus et al. (2024a), further amplifying our ability to detect membership.

3.2 Detecting Dataset Membership

To detect membership, we leverage the insight that under the null hypothesis $\boldsymbol{H_0}$ (no membership), the model should not exhibit any systematic preference towards any of the semantically equivalent paraphrases of documents that are watermarked with distinct keys—the public version of the dataset and privately held versions of the dataset. This follows from the randomness inherent in our selection of keys and nature of watermark we employ. We formalize this intuition through a statistical testing framework.

For each document q, we compute the perplexity difference d_i between its public version $q^{'}_i$ and private version $q^{''}_i$ that form a pair $(q^{'}_i, q^{''}_i)$:

$$d_i = \mathit{PPL}_\theta(q^{'}_i) - \mathit{PPL}_\theta(q^{''}_i). \tag{2}$$

Prior to applying the paired t-test, we modify the top 5% outliers by clipping their values. This prevents issues where the test can become ineffective due to a few outlier samples. Under the alternative hypothesis $\boldsymbol{H_1}$, we expect these differences to be negative on average, indicating lower perplexity for public versions. We evaluate this using a one-sided paired t-test statistic:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}, \tag{3}$$

where $\bar{d}$ and s_d are the mean and standard deviation of the differences across the collection of documents respectively and n is the number of documents (Student, 1908). The one-sided p-value specifically tests for $\bar{d} < 0$, following our alternative hypothesis that exposure during training leads to lower perplexity on public versions. Paired tests provide higher statistical power, enabling detecting membership even with a smaller collection of documents (n), as we show in our experiments.

Multiple Private Keys. In practice, we empirically observe that models may exhibit inherent biases at an individual document level, occasionally assigning lower perplexity to specific private rephrases (q'') independent of membership. To make our detection robust against such biases, we propose using multiple private rephrases, each watermarked with a distinct key. Instead of comparing against a single private version, we test whether public rephrases exhibit lower perplexity compared to the average perplexity across m different private rephrases:

$$d_i = \mathit{PPL}_\theta(q^{'}_i) - \frac{1}{m}\sum_{j=1}^{j=m} \mathit{PPL}_\theta(q^{''}_{i,j}), \tag{4}$$

[2]The public and private keys are chosen randomly, with one key designated as the public key.

Table 1: **P-values for detecting *test-set contamination* for different methods.** For LLM DI (Maini et al., 2024), *same* refers to using rephrases of the benchmark questions as *validation* set, while *different* uses an entirely different set of unseen questions from the same benchmark as the *validation* set. **Bold** indicates statistically significant results ($p < 0.05$). Across all the four benchmarks, our approach results in lower p-values compared to other approaches (lower is better).

	Benchmark			
Method	TriviaQA	Arc-C	MMLU	GSM8K
PaCoST (Zhang et al., 2024a)	**1.6e-3**	0.33	0.19	0.21
LLM DI (Maini et al., 2024) (*same*)	0.43	0.31	0.46	0.30
LLM DI (Maini et al., 2024) (*different*)	**0.02**	0.53	**0.03**	0.71
STAMP (w/o paired tests)	0.14	0.07	0.08	**0.02**
STAMP (w/o watermarking)	**0.02**	**5.1e-3**	**0.02**	**1.4e-3**
STAMP	**1.2e-4**	**2.8e-4**	**7.0e-4**	**6.6e-6**

where m is a hyperparameter known as *private key count*, and $q''_{i,j}$ represents the j^{th} private rephrase of i^{th} document. Through controlled experiments, we analyze the effect of this hyperparameter on statistical strength of our test (§4.4).

4 Experiments & Results

To evaluate the ability of **STAMP** for membership detection, we first focus on benchmark contamination—the inclusion of evaluation benchmarks in the pretraining corpora of LLMs. This setting presents unique challenges for membership detection. First, benchmarks must maintain their utility as reliable indicators of progress, which constrains the modifications we can make prior to their release. Second, benchmarks typically contain limited text compared to other content types (e.g., books or newsletters), making detection particularly challenging.

4.1 Releasing Watermarked Test Sets

We evaluate our approach using four widely-used benchmarks: TriviaQA Joshi et al. (2017), ARC-C Clark et al. (2018), MMLU Hendrycks et al. (2020), and GSM8K Cobbe et al. (2021). For each benchmark, we follow our proposed methodology (§3.1) to generate watermarked public and private paraphrases. We use the instruction tuned Llama3-70B AI@Meta (2024) model and a benchmark-agnostic prompt (provided in Appendix J) to generate these rephrased copies. For each benchmark, we randomly select one watermarked version to be the *public* version. Examples of the rephrased test instances are provided in Appendix K.

Key Distinction. While rephrasing has been previously explored for detecting contamination Zhang et al. (2024a), existing approaches typically compare human-written content against their LLM-generated rephrases, overlooking a crucial confounding factor: language models exhibit systematic preferences for LLM-generated text over human-written content Liu et al. (2023a); Mishra et al. (2023); Laurito et al. (2024). This inherent bias undermines the reliability of statistical approaches that compare human-written content with their LLM rephrasings, as any detected differences might stem from this general preference rather than training exposure. To enable reliable statistical testing, it is crucial to control the data generating process for both versions being compared. We address this by ensuring both our public and private versions are generated through the same process, differing only in their watermarking keys. Given the random selection of keys, we expect no systematic preferences between versions unless one was seen during training.

We empirically validate that human-written content and its LLM-generated rephrasings are easily distinguishable (thus violating the expected IID requirement): a simple bag-of-words classifier obtains AUROC > 0.8 on four out of five benchmarks, whereas the classifier performs no better than

Table 2: **Performance of models on the original datasets compared to the watermarked benchmarks.** We evaluate the models using the LM evaluation harness Gao et al. (2024) with the default settings, comparing performance on original benchmarks against two watermarking approaches: UNICODE substitutions Wei et al. (2024) and **STAMP**. Due to space constraints, results for MMLU and GSM8K benchmarks are presented in Table X of Appendix A. We find that models obtain comparable performance on **STAMP**-watermarked benchmarks, but crucially, **the relative ranking of LLMs remains unchanged across all benchmarks.**

Dataset	Metric	Variant	Pythia 1B	Gemma-2 2B	Mistral 7B	LLaMA-3 8B	Gemma-2 9B
ARC-C	0-shot	Original	26.1	48	49.1	50.6	59.0
		Unicode	21.6	37.3	39.0	41.5	49.8
		STAMP	26.3	46.8	49.1	50.5	57.1
TriviaQA	5-shot	Original	12.4	52.7	67.2	68.9	70.1
		Unicode	1.1	23.6	46.0	44.3	54.8
		STAMP	11.4	51.9	65.9	66.3	68.6

random chance when distinguishing between rephrasings watermarked with different keys. Detailed analysis and classifier specifications are provided in Appendix C.

4.2 PRETRAINING WITH INTENTIONAL CONTAMINATION

Setup. To simulate downstream benchmark contamination as it occurs in real-world scenarios and evaluate the effectiveness of our test, we perform continual pretraining on the 1 billion parameter Pythia model Biderman et al. (2023) using an intentionally contaminated pretraining corpus. The corpus is a combination of OpenWebText Contributors (2023) and *public* watermarked version of the four benchmarks, as mentioned in Section 4.1. Each test set accounts for less than **0.001%** of the pretraining corpus, with exact sizes detailed in Table 6 in the appendix. All test sets in our experiments have a duplication rate of 1 (denoting no duplication whatsoever), and the overall pretraining dataset comprises 6.7 billion tokens. Details of the exact training hyperparameters are provided in Appendix E.

Baselines. We compare **STAMP** against two recent statistical approaches to detect membership: PaCoST Zhang et al. (2024a) and LLM DI Maini et al. (2024). PaCoST employs a paired t-test that compares model confidence on original and rephrased versions, while LLM DI aggregates multiple membership inference attacks (MIAs) to perform statistical testing. For LLM DI, which requires access to an unseen *validation* set, we evaluate two settings: (1) using private rephrases of the publicly available dataset as the *validation* set, and (2) using an entirely different set of documents from the same distribution as the *validation* set.

Additionally, we also evaluate state-of-the-art MIAs: *PPL* Yeom et al. (2018), *Zlib* Carlini et al. (2021), *Min-K* Shi et al. (2024), *Min-K++* Zhang et al. (2024b) and *DC-PDD* Zhang et al. (2024c). Since MIAs rely on a non-trivial detection threshold, we report AUROC scores across two settings: (1) discriminating between public rephrases in training and private rephrases of the same documents, and (2) discriminating between public rephrases in training and unseen documents from the same dataset.

Main Results. We compare **STAMP** and baseline methods in Table 1. **STAMP** achieves statistically significantly low p-values (ranging from 10^{-4} to 10^{-6}) across all benchmarks, substantially outperforming existing methods. In contrast, PaCoST detects contamination only on TriviaQA ($p \approx 10^{-3}$), while LLM DI shows significance on just two benchmarks (TriviaQA and MMLU) even with access to validation data of extra test examples.

In our experiments, all MIA methods achieve an AUROC score of ≈ 0.5 across all benchmarks, indicating performance no better than random guessing. Detailed MIA results and analysis are presented in Table 8.

Table 3: **False positive analysis.** P-values obtained using **STAMP** for both an uncontaminated and contaminated model to detect membership of watermarked benchmarks *excluded from the pretraining corpora*. High p-values denote that our approach does not falsely detect membership.

Dataset	Pythia Uncontaminated	Pythia Contaminated
TriviaQA	0.52	0.28
ARC-C	0.31	0.56
MMLU	0.54	0.15
GSM8K	0.38	0.47
Paper Abstracts	0.55	0.07
Blog Articles	0.21	0.73

False Positive Analysis. To ensure the robustness of **STAMP** against false positives, we conduct two key experiments. First, we apply our detection methodology to off-the-shelf pretrained LLMs that have not been exposed to the watermarked benchmarks. The results for Pythia 1B, presented in the first column of Table 3, show no false positives. We extend this analysis to models of different sizes and families in Table **??**, consistently finding no false positives across all tested models, confirming the robustness of **STAMP** against false positives. Second, we perform a stronger test to evaluate whether **STAMP** detects the membership of the dataset rather than just distributional differences due to different watermarking keys. We create held-out subsets from the same benchmarks and watermark them using the identical public keys used for our contaminated versions. While these held-out sets share the same distribution and watermarking as our training data, they contain entirely different examples. We then apply our detection methodology to test if these *held-out* sets are falsely detected as members in our contaminated Pythia 1B model. The second column of Table 3 shows consistently large p-values, indicating **STAMP** successfully refutes membership for these *held-out* sets.

Performance Without Watermarks Embedded. To validate our hypothesis that using a watermarked LLM to generate the rephrased copies of the benchmark enhances the statistical strength of our test, we conduct experiments under the same settings as described above (§4.2), but with rephrased copies generated without using a watermarked LLM. The results, presented in Table 1, confirm that incorporating watermarked test sets significantly boosts the statistical power of our test, improving performance by at least two orders of magnitude across all benchmarks.

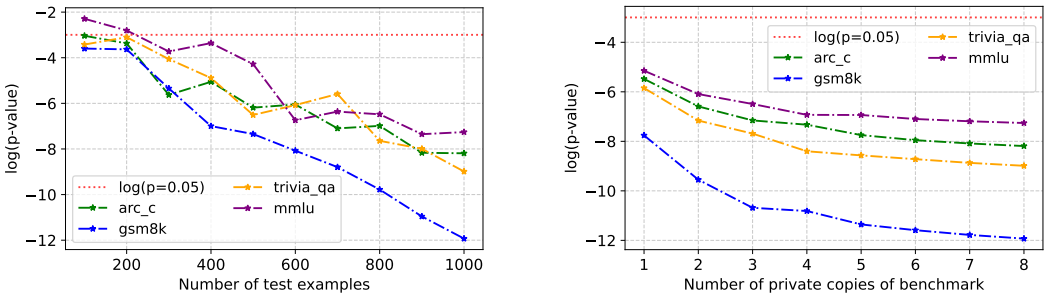
4.3 Utility of Test Sets

Detecting contamination alone is insufficient; the watermarked content should retain the desired properties (for e.g., benchmarks should maintain their utility as reliable indicators of LLM performance). Using the lm-evaluation-harness framework Gao et al. (2024), we assess five pre-trained LLMs on both original and watermarked benchmarks. Additionally, we measure semantic preservation using the P-SP metric Wieting et al. (2021).

Our results, presented in Table 2, demonstrate that **STAMP**-watermarked variants maintain benchmark utility: LLMs achieve similar absolute performance and the relative rankings of LLMs across all benchmarks are unaffected. In contrast, UNICODE watermark Wei et al. (2024) significantly degrades benchmark utility, with performance drops of up to 20% and does not preserve relative rankings. **STAMP**-watermarked variants also result in high semantic preservation (P-SP scores between 0.83 & 0.91) across all benchmarks. For reference, the average score of human paraphrases is 0.76 as per Krishna et al. (2024). These results are available in Table 5.

4.4 Parameters Affecting the Power of the Test

Benchmark size. To analyze the effect of sample size (n) on detection power, we evaluate our test on benchmark subsets ranging from 100 to 1000 examples. For each size, we average p-values across 10 runs with different random seeds. Our results, in Figure 2a, demonstrate that our approach



(a) Log p-value vs sample size of benchmark (n)

(b) Log p-value vs private key count (m)

Figure 2: **Impact of benchmark size (n) and private key count (m) on STAMP's statistical power.** The dotted red line indicates the standard significance threshold ($p = 0.05$). Lower values indicate stronger statistical evidence of contamination.

works even with just 600 examples, where we consistently achieve low p-values ($\approx 10^{-3}$) across all datasets.

Private key count. Our proposed test compares the perplexity of the public version against the average perplexity of m private versions (Equation 4). Here we analyze how this hyperparameter (m) affects the statistical power of our test. As shown in Figure 2b, increasing the number of private keys strengthens detection up to a threshold of 5 keys, beyond which we see negligible improvement.

Size of Pretraining Corpora. We analyze our test's effectiveness for different scales of pretraining data by combining contaminated benchmarks with varying amounts of OpenWebText data Contributors (2023). We note that while the strength decreases with corpus size, the *rate of decline* diminishes substantially beyond 4 billion tokens, with minimal drop in detection strength between 4 and 6 billion tokens (Figure 3). Notably, these results are obtained with a modest 1B-parameter model; given that larger models exhibit stronger memorization Carlini et al. (2019), we believe that **STAMP** will detect membership for larger models.

5 Real World Case Studies

To demonstrate **STAMP**'s effectiveness in detecting *unlicensed* use of copyrighted data in model training, we present two expository case studies. Specifically, we apply **STAMP** to detect membership of (1) abstracts from EMNLP 2024 proceedings Al-Onaizan et al. (2024) and (2) articles from the AI Snake Oil newsletter Narayanan & Kapoor (2023).

Paper Abstracts. We sample 500 papers from EMNLP 2024 proceedings Al-Onaizan et al. (2024) and generate watermarked rephrasings of their abstracts. Additionally, we generate watermarked rephrasings for another set of 500 abstracts, which we use as a *held-out* validation set for our experiments. The prompt templates used for rephrasing and examples of watermarked abstracts are provided in Appendix J and Appendix K , respectively.

To evaluate whether the semantic content of abstracts is preserved, we use the P-SP metric (Wieting et al., 2021), where watermarked abstracts achieve a high score of 0.95, indicating that the semantic content is largely preserved. To further evaluate the acceptability of watermarked abstracts, we conduct both an automated evaluation (using GPT-4) and a small-scale human study involving original authors. In both evaluations, participants compare the original abstract and its watermarked rephrasing, classifying the latter into one of five options: *preferred*, *acceptable*, acceptable with *minor revisions*, or *major revisions*, and lastly *unacceptable*. Detailed evaluation protocols are provided in Appendix H.

For 1000 watermarked abstracts, 99% were rated by GPT-4 as either *preferred* or *acceptable*. In a preliminary human study, we ask authors to review rephrasings of their own abstracts. Of the 40 watermarked abstracts evaluated, authors find 24 to be acceptable as is, indicate 11 could use

Table 4: **Case studies.** We report p-values of different approaches for detecting dataset membership (lower is better). LLM DI (same) uses the private rephrasing of the same documents, while LLM DI (different) uses different documents from a held out set from the same distribution.

Method	Paper Abstracts	Blog Articles
LLM DI (same)	0.15	0.44
LLM DI (different)	**0.05**	0.58
STAMP (w/o paired tests)	**0.01**	0.07
STAMP	**2.7e-12**	**2.4e-3**

minor edits, and 4 prefer the rephrased version over their self-written abstracts, with just 1 abstract requiring major edits. Details of the evaluation are provided in Appendix H.

Blog Posts from AI Newsletter. We collect 56 posts from the popular AI Snake Oil newsletter Narayanan & Kapoor (2023), and use 44 for pretraining and hold 12 for validation. To demonstrate how **STAMP** could handle longer-form content, we adapt it to rephrase at the paragraph level, treating each paragraph as an independent datapoint for our test. Note, while each paragraph serves as a datapoint for our test, the blog posts are included in the pretraining corpora at the document level, following standard pretraining practices (detailed in Appendix E). We present the prompt used to rephrase in Appendix J

We evaluate **STAMP**'s ability to detect dataset membership by performing continual pretraining on the Pythia 1B model using a training corpus composed of watermarked paper abstracts ($\approx$ 105K tokens), watermarked blog posts ($\approx$ 95K tokens), and a subset of OpenWebText ($\approx$ 3.3B tokens). Additionally, to verify that **STAMP** can detect dataset membership for distinct datasets watermarked with the same key, we apply a consistent watermarking key when generating the public versions of both datasets.

Results. Our results in Table 4 demonstrate that **STAMP** effectively detects dataset membership for both paper abstracts and blog posts, achieving statistically significant p-values. To compare, we evaluate LLM DI under different choices of validation set: first, using private rephrases of the same documents and second, using a different held-out set of documents watermarked with the same public key. While LLM DI can detect membership for paper abstracts, it fails to do so for blog posts. Further, membership inference attacks exhibit near-random performance (Table 8). To verify the robustness of **STAMP** against false positives, we evaluate it under the two settings discussed earlier (§4.2). Our results in Table 3 confirm that **STAMP** does not result in any false positives, reinforcing its reliability.

6 Related Work

We discuss the most closely related work below, focusing on statistical approaches for detecting dataset membership, test-set contamination and the use of watermarks for detecting membership of datasets. A more comprehensive review of related literature is provided in Appendix G.

Dataset Membership. A recent hypothesis-testing approach embeds random sequences in text or substitutes characters with visually-similar unicodes Wei et al. (2024). Similarly, Meeus et al. (2024a) propose inserting "copyright traps" into documents to enhance document-level membership inference. These methods then test the model's preference for these inserted sequences or substitutions. However, such alterations impair machine readability, making them impractical for content creators. Another recent proposal Maini et al. (2024) selectively combines membership inference attacks (MIAs) that provide positive signals for a given distribution and aggregates them to perform a statistical test on a dataset. Their method assumes access to a validation set drawn from the same distribution as the target dataset and unseen by the model—a difficult requirement to satisfy.

A recent position paper (Zhang et al., 2025) argues that methods attempting to estimate FPR by collecting non-members a posteriori are statistically unsound, a position that aligns with our analysis of

PaCoST (Zhang et al., 2024a). We believe **STAMP** aligns with the criteria for a sound membership proof presented in the paper. Specifically, we use private members sampled from the same distribution as the publicly released version x. Since our private members are semantically equivalent to the public member, any causal effects of publishing x would similarly affect the private members, ensuring the statistical validity of our approach.

Test Set Contamination. While our focus is detecting membership of any arbitrary collection of documents, some recent *statistical approaches* have focused on detecting test set contamination. A recent work Oren et al. (2023) proposes a permutation test based on the canonical ordering in a benchmark but relies on the strong assumption of metadata contamination (canonical ordering of the dataset). Another recent proposal Zhang et al. (2024a) compares the model confidence on test instances and their rephrased counterparts. However, as discussed earlier (§4.1), LLMs may favor their own outputs, and this is an oft-overlooked confounder. Additionally, there have been a few approaches based on prompting models to reproduce near-exact test examples Sainz et al. (2023b); Golchin & Surdeanu (2024). However, the heuristic-y nature of these approaches prevents them from providing statistical evidence of contamination.

Watermarking for Dataset Membership. A few recent approaches have explored using LLM watermarks for membership detection. Waterfall (Lau et al., 2024) proposes a watermarking scheme for protecting IP of text and further demonstrates how to detect unauthorized fine-tuning of LLMs on proprietary text data. Specifically, to detect membership of a text, their approach prompts the target model with a prefix and detects the embedded watermarking in the generated new tokens to test for membership. While effective in certain scenarios, their approach requires a higher level of memorization and has only been demonstrated in fine-tuning settings with multiple epochs. Additionally, their method is not applicable to domains like benchmarks where each sample is only a few tokens long. These limitations may restrict its practical utility for detecting membership of a dataset in the pretraining corpora of an LLM.

Another recent contemporaneous study (Sander et al., 2025) proposes a similar approach where watermarks are embedded in benchmarks by reformulating the original questions with a watermarked LLM. While employing a similar setup, their detection approach differs substantially from ours. Their method relies on detecting overfitting of the contaminated model on token-level watermarking biases to prove contamination, whereas our approach compares perplexity differences between the publicly released benchmarks and private versions watermarked with different keys.

7 Conclusion & Future Directions

In this work, we presented **STAMP**, a statistical framework for detecting dataset membership, which can reliably be used by content creators to watermark their content, while preserving the utility, or the meaning, of the original content. We demonstrated **STAMP**'s effectiveness in detecting test-set contamination through comprehensive experiments. Our ablation studies systematically analyzed how detection strength varies with dataset size, the number of private versions, and pretraining corpus size. We validated the real-world applicability of our approach through two case studies: detecting paper abstracts and blog posts in pretraining data.

There are several **important limitations** of our work: first, watermarks must be embedded before the content is released online, making it inapplicable to already published content. We believe this is a fundamental limitation shared by existing statistical methods, as they require knowledge of the data-generating process to construct a valid null distribution. Finally, due to computational constraints, we evaluated our approach using continual pretraining rather than training models from scratch. While our results demonstrate effectiveness in this setting outperforming baselines, future work could validate these findings using models that are trained from scratch.

Acknowledgments

We sincerely thank all participants in our evaluation study for their valuable time. This work was supported in part by the AI2050 program at Schmidt Sciences (Grant G-24-66186). Additionally, DP is grateful to Adobe Inc., Pratiksha Trust and the National Payments Corporation of India (NPCI) for generously supporting his group's research.

REFERENCES

AI@Meta. Llama 3 model card, 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.). *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.0. URL https://aclanthology.org/2024.emnlp-main.0/.

Anthropic. Claude 3 model card, 2024. URL https://www.anthropic.com/news/claude-3-family.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023. URL https://arxiv.org/abs/2304.01373.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pp. 267–284, 2019.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models, 2023.

Debeshee Das, Jie Zhang, and Florian Tramèr. Blind baselines beat membership inference attacks for foundation models, 2024. URL https://arxiv.org/abs/2406.16201.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*, 2024.

Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hanna Hajishirzi, Noah A. Smith, and Jesse Dodge. What's in my big data?, 2024. URL https://arxiv.org/abs/2310.20707.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Shahriar Golchin and Mihai Surdeanu. Time travel in llms: Tracing data contamination in large language models, 2024. URL https://arxiv.org/abs/2308.08493.

Michael M. Grynbaum and Ryan Mac. The times sues openai and microsoft over a.i. use of copyrighted work https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html, 2023a. URL https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html.

Michael M. Grynbaum and Ryan Mac. Sarah silverman and authors sue openai and meta over copyright infringement, 2023b. URL https://www.nytimes.com/2023/07/10/arts/sarah-silverman-lawsuit-openai-meta.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks, 2023. URL https://arxiv.org/abs/2305.10160.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models, 2024. URL https://arxiv.org/abs/2306.04634.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36, 2024.

Gregory Kang Ruey Lau, Xinyuan Niu, Hieu Dao, Jiangwei Chen, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Waterfall: Framework for robust and scalable text watermarking and provenance for llms, 2024. URL https://arxiv.org/abs/2407.04411.

Walter Laurito, Benjamin Davis, Peli Grietzer, Tomáš Gavenčiak, Ada Böhm, and Jan Kulveit. Ai ai bias: Large language models favor their own generated content. *ArXiv*, abs/2407.12856, 2024. URL https://api.semanticscholar.org/CorpusID:271270236.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment, 2023a. URL https://arxiv.org/abs/2303.16634.

Yixin Liu, Hongsheng Hu, Xun Chen, Xuyun Zhang, and Lichao Sun. Watermarking text data on large language models for dataset copyright. *arXiv preprint arXiv:2305.13257*, 2023b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 157–165, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.18. URL https://aclanthology.org/2022.acl-short.18.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. *arXiv preprint arXiv:2104.10706*, 2021.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset?, 2024. URL https://arxiv.org/abs/2406.06443.

Matthieu Meeus, Igor Shilov, Manuel Faysse, and Yves-Alexandre De Montjoye. Copyright traps for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. Sok: Membership inference attacks on llms are rushing nowhere (and how to fix it). 2024b. URL https://api.semanticscholar.org/CorpusID:270737841.

Aditi Mishra, Sajjadur Rahman, Han Jun Kim, Kushan Mitra, and Estevam R. Hruschka. Characterizing large language models as rationalizers of knowledge-intensive tasks. *ArXiv*, abs/2311.05085, 2023. URL https://api.semanticscholar.org/CorpusID:265067226.

Arvind Narayanan and Sayash Kapoor. Ai snake oil., 2023. URL https://www.aisnakeoil.com/. Newsletter.

NewYorkTimes. The times sues openai and microsoft over a.i. use of copyrighted work. https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html, 2023.

OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto. Proving test set contamination in black box language models, 2023. URL https://arxiv.org/abs/2310.17623.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.722. URL https://aclanthology.org/2023.findings-emnlp.722.

Oscar Sainz, Jon Ander Campos, Iker García-Ferreroa, and Julen Etxaniz andEneko Agirre. Did chatgpt cheat on your test? https://hitz-zentroa.github.io/lm-contamination/blog/, 2023b.

Tom Sander, Pierre Fernandez, Saeed Mahloujifar, Alain Durmus, and Chuan Guo. Detecting benchmark contamination through watermarking. *arXiv preprint arXiv:2502.17259*, 2025.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models, 2024. URL https://arxiv.org/abs/2310.16789.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models, 2017. URL https://arxiv.org/abs/1610.05820.

Aaditya K Singh, Muhammed Yusuf Kocyigit, Andrew Poulton, David Esiobu, Maria Lomeli, Gergely Szilvasy, and Dieuwke Hupkes. Evaluation data contamination in llms: how do we measure it and (when) does it matter? *arXiv preprint arXiv:2411.03923*, 2024.

Student. The probable error of a mean. *Biometrika*, pp. 1–25, 1908.

Liyan Tang, Philippe Laban, and Greg Durrett. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8818–8847, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.499. URL https://aclanthology.org/2024.emnlp-main.499/.

Katherine Thai, Marzena Karpinska, Kalpesh Krishna, Bill Ray, Moira Inghilleri, John Wieting, and Mohit Iyyer. Exploring document-level literary machine translation with parallel paragraphs from world literature. *arXiv preprint arXiv:2210.14250*, 2022.

Johnny Tian-Zheng Wei, Ryan Yixiang Wang, and Robin Jia. Proving membership in llm pretraining data via data watermarks. *arXiv preprint arXiv:2402.10892*, 2024.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. Paraphrastic representations at scale. *arXiv preprint arXiv:2104.15114*, 2021.

Zhijun Xu, Siyu Yuan, Lingjie Chen, and Deqing Yang. "a good pun is its own reword": Can large language models understand puns? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11766–11782, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.657. URL https://aclanthology.org/2024.emnlp-main.657/.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282. IEEE, 2018.

Armel Zebaze, Benoît Sagot, and Rachel Bawden. Tree of problems: Improving structured problem solving with compositionality, 2024. URL https://arxiv.org/abs/2410.06634.

Huixuan Zhang, Yun Lin, and Xiaojun Wan. Pacost: Paired confidence significance testing for benchmark contamination detection in large language models, 2024a. URL https://arxiv.org/abs/2406.18326.

Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. Membership inference attacks cannot prove that a model was trained on your data, 2025. URL https://arxiv.org/abs/2409.19798.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-k%++: Improved baseline for detecting pre-training data from large language models. *arXiv preprint arXiv:2404.02936*, 2024b.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. Pretraining data detection for large language models: A divergence-based calibration method, 2024c. URL https://arxiv.org/abs/2409.14781.

A ADDITIONAL RESULTS

Table 5: **Semantic similarity scores** (P-SP) (Wieting et al., 2021) between original datasets and their watermarked rephrases. For reference: the P-SP value is 0.76 for human-written paraphrases as per a recent study (Krishna et al., 2024).

	TriviaQA	ARC-C	MMLU	GSM8K	Paper Abstracts
P-SP	0.91	0.83	0.86	0.90	0.95

Table 6: **Size of evaluation benchmark used in the intentional contamination experiment (§4.4).** Each benchmark is subsampled to 1,000 examples, with each injected benchmark making up less than 0.001% of the entire pretraining corpus, which consists of 6.7 billion tokens. Each benchmark is injected exactly once into the corpus without any duplication.

BENCHMARK	SIZE (TOKENS)	% PRETRAINING DATA
TRIVIAQA	34609	5.1E-4
ARC-C	36863	5.5E-4
MMLU	42548	6.3E-4
GSM8K	61132	9.0E-4

Table 7: **Performance of models on the original datasets compared to the watermarked benchmarks.** We evaluate the models using the LM evaluation harness Gao et al. (2024) with the default settings, comparing performance on original benchmarks against two watermarking approaches: UNICODE substitutions Wei et al. (2024) and **STAMP**. Due to space constraints, results for MMLU and GSM8K benchmarks are presented in Table X of Appendix A. We find that models obtain comparable performance on **STAMP**-watermarked benchmarks, but crucially, **the relative ranking of LLMs remains unchanged across all benchmarks.**

Dataset	Metric	Variant	Pythia 1B	Gemma-2 2B	Mistral 7B	LLaMA-3 8B	Gemma-2 9B
ARC-C	0-shot	Original	26.1	48	49.1	50.6	59.0
		Unicode	21.6	37.3	39.0	41.5	49.8
		STAMP	26.3	46.8	49.1	50.5	57.1
MMLU	5-shot	Original	28.1	52.9	59	61.1	68.6
		Unicode	28.4	45.0	51.5	55.9	63.2
		STAMP	28.8	51.6	56	61.8	68.4
TriviaQA	5-shot	Original	12.4	52.7	67.2	68.9	70.1
		Unicode	1.1	23.6	46.0	44.3	54.8
		STAMP	11.4	51.9	65.9	66.3	68.6
GSM8K	5-shot	Original	1.6	25.8	34.4	51.8	65.5
		Unicode	1.5	23.1	23.3	46.7	60.8
		STAMP	2.2	27.2	37.5	54.9	65.8

A.1 DETECTING PARTIAL CONTAMINATION

In practice, benchmarks may be partially contaminated, where only a subset of test examples appears in the pretraining corpora. Understanding the impact of partial contamination is critical because benchmark owners cannot identify which specific test examples have been leaked. This study

Table 8: **Comparison of Membership Inference Attacks (MIA) performance across different datasets.** We report AUC scores for three MIA methods under two settings: *Same Documents:* public rephrases in training vs private rephrases of the same documents, and *Different Documents:* public rephrases in training vs different unseen documents from the same dataset. AUROC score of ≈ 0.5 indicates performance no better than random guessing.

	Same Documents. (↑)					*Different Documents.* (↑)				
DATASET	PPL	ZLIB	MIN-K	MIN-K++	DC-PDD	PPL	ZLIB	MIN-K	MIN-K++	DC-PDD
TRIVIAQA	0.50	0.50	0.50	0.50	0.52	0.46	0.57	0.48	0.44	0.58
ARC-C	0.50	0.50	0.50	0.49	0.51	0.49	0.50	0.48	0.45	0.52
MMLU	0.48	0.49	0.48	0.49	0.52	0.43	0.48	0.44	0.45	0.52
GSM8K	0.50	0.50	0.50	0.50	0.52	0.47	0.47	0.48	0.48	0.52
PAPER ABSTRACTS	0.48	0.49	0.48	0.46	0.53	0.41	0.46	0.42	0.40	0.55
BLOG ARTICLES	0.50	0.51	0.51	0.49	0.50	0.49	0.51	0.48	0.46	0.51

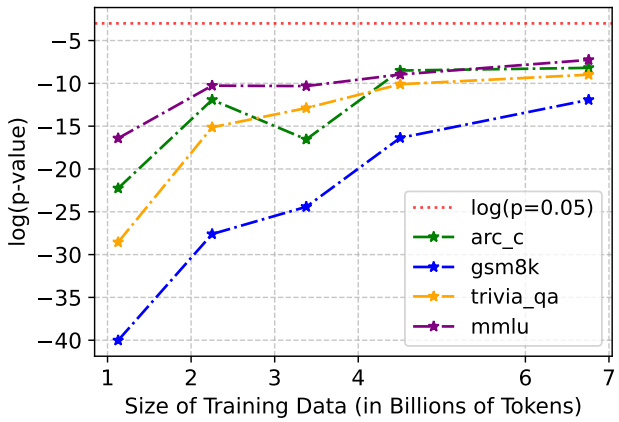


Figure 3: **Log p-value vs pretraining corpus size.** We observe that the *rate of decline* diminishes as we increase the corpus size, with negligible drop between 4B and 6B.

complements our earlier analysis in Section 4.4, by focusing on the sensitivity of our approach under varying proportions (α) of contaminated examples within a fixed benchmark size (n).

Our results in Figure 4 highlight that as α increases the detection strength improves, with p-values dropping below 10^{-3} when majority of the benchmark is contaminated. We also observe that **STAMP** reliably detects contaminated even when only 40% of the test examples are contaminated. Our findings confirm that **STAMP** successfully identifies contamination with high statistical significance, even in scenarios of partial contamination.

B P-SP METRIC

To validate semantic preservation in our watermarking process, we employ P-SP Wieting et al. (2021), a state-of-the-art semantic similarity model. P-SP uses embedding averaging trained on a large corpus of filtered paraphrase data, and has been shown to effectively distinguish between true paraphrases and unrelated text. As evidenced by Krishna et al. (2024), P-SP assigns an average score of 0.76 to human-created paraphrases in the PAR3 dataset Thai et al. (2022), while random paragraph pairs from the same book score only 0.09. Table 5 reports the average P-SP scores between original benchmarks and their watermarked versions across 9 random *hash keys*. Our watermarked versions achieve high P-SP scores (0.83-0.95) across all benchmarks, substantially exceeding the average score for human paraphrases, indicating strong semantic preservation.

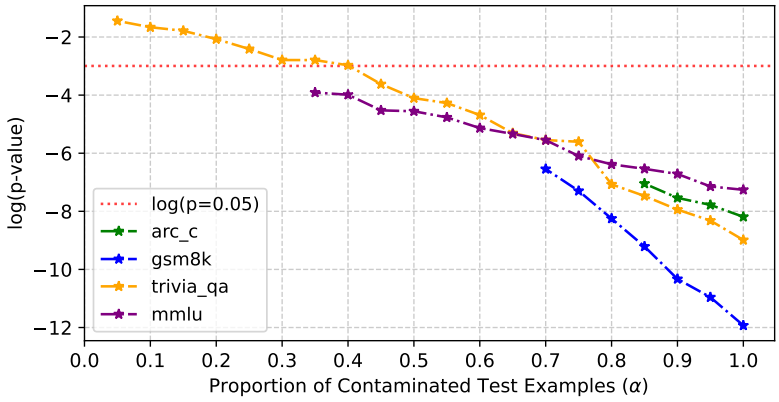


Figure 4: **Log p-value vs proportion of benchmark that is contaminated**. We plot the log p-value against the proportion of test examples that are leaked to analyze the sensitivity of our test to detect contaminated in scenarios where the benchmark is only partially contaminated.

C Bag-of-Words Classifier

We train a random forest classifier on the *bag-of-words* feature representations for the datasets. The classifier is trained on 80% of the member and non-member sets, with evaluation performed on the remaining 20%. Results are aggregated over a 5-fold cross-validation. The detailed results are presented in Table 9.

Table 9: **AUROC using *bag-of-words* features to distinguish between different versions of datasets.** The first column shows AUROC for distinguishing original datasets from their rephrased versions, where high values (> 0.8) indicate clear distributional differences. The second column shows AUROC for distinguishing between *public* and *private* watermarked versions, where values near 0.5 indicate distributional similarity.

Dataset	Original vs Rephrased	Public vs Private
TriviaQA	0.66	0.51
Arc-C	0.83	0.52
MMLU	0.83	0.53
GSM8k	0.84	0.57
paper abstracts	0.86	0.57

D Perplexity

Perplexity (*PPL*) measures how well a language model predicts a given text sequence S, with lower values indicating better prediction. For an auto-regressive language model θ and text sequence S, tokenized as a sequence of N tokens $\{s_1, \ldots, s_N\}$, perplexity is computed as the exponent of the loss. Formally:

$$PPL_\theta(S) = \exp\left(\mathcal{L}_\theta(S)\right) \tag{5}$$

Where the loss $\mathcal{L}_\theta$ is defined as:

$$\mathcal{L}_\theta(S) = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\mathcal{P}_\theta(s_i|s_{<i})\right) \tag{6}$$

Here $\mathcal{P}_\theta(s_i|s_{<i})$ denotes the predicted probability for token s_i by the language model θ given the context of previous tokens $\{s_1, \ldots, s_{i-1}\}$.

E PRETRAINING DETAILS

We continually pretrain Pythia 1B on a mixture of OpenWebText and the evaluation benchmarks. Test case instances from the benchmark were randomly inserted between documents from OpenWebText. We trained for 1 epoch of 46000 steps with an effective batch size of 144 sequences and sequence length of 1024 tokens. We used the AdamW optimizer Loshchilov & Hutter (2019) with a learning rate of 10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$ and no weight decay.

F WATERMARK FOR LARGE LANGUAGE MODELS

In work, we use the prominent KGW Kirchenbauer et al. (2023) watermarking scheme. KGW scheme uses a hash function that takes the context (preceding tokens) and a hash key h to partition the vocabulary V into two disjoint sets at each generation step: a green list G and a red list R. Formally, for a language model $\mathcal{M}$ with vocabulary V, and a prefix comprising tokens $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$, the scheme involves first computing the logits $\mathcal{M}(\mathbf{w}_1 \ldots, \mathbf{w}_n) = (l_1, \ldots, l_{|V|})$ of the language model that would ordinarily be used to predict the subsequent token. As per a hyper-parameter k, the last k tokens, $\mathbf{w}_{n-k+1}$ to $\mathbf{w}_n$, are then fed to a pseudo-random function F to partition V into a green list G and a red list R such that $|G|+|R| = |V|$. Finally, the logits corresponding to the tokens in the green list, G, are boosted by δ ($\delta > 0$). Specifically, in our work we set $k = 1$ and $\delta = 1.0$ as the chose hyperparameters. The watermark can then be detected through a one-proportion z-test on the fraction of green tokens in the generated text.

G RELATED WORKS

Our works relates to a large literature of work on membership inference (§G.2), dataset membership (§G.3) and test-set contamination detection (§G.4) in large language models.

G.1 WATERMARKING FOR DATASET MEMBERSHIP

Liu et al. (2023b) propose TextMarker, a backdoor-based membership inference technique for protection of *classification datasets*. Their approach watermarks each original sample (x, y) by inserting specific triggers (character or word-level substitutions) into x, creating a backdoored sample x_t. They then assign an altered target label y_t ($y_t \neq y$) to this modified input. Membership detection is performed by testing whether a model f produces the watermarked label with high probability: $\Pr(f(x_t) = y_t)$. While effective, TextMarker is specifically designed for classification datasets and it is not trivial to extend it to other kinds of benchmarks or the broader problem of dataset membership detection, limiting its real-world applicability.

Recent work has proposed detecting watermark signals in suspect model outputs to determine dataset membership. Waterfall (Lau et al., 2024) enables creators to watermark their text using a robust watermarking scheme that leverages an LLM for rephrasing. They further demonstrate that watermarks in fine-tuning data, persist in downstream LLM outputs, allowing for membership detection. While both approaches use LLM-based rephrasing, Waterfall relies on detecting watermarks directly in model generations, whereas our method uses multiple watermarked versions to detect perplexity divergences. While Waterfall is effective for text watermarking, it has limited utility for pretraining membership detection. It relies on strong overfitting and presumes that the model is trained on datasets over multiple epochs. Additionally, their approach is not applicable to short text segments commonly found in benchmarks. Another recent contemporaneous study (Sander et al., 2025) proposes a similar approach where watermarks are embedded in benchmarks by reformulating the original questions with a watermarked LLM. While employing a similar setup, their detection approach differs substantially from ours. Their method relies on detecting model overfitting on the green tokens in the watermarked benchmark to prove contamination, whereas our approach compares perplexity differences between the publicly released benchmarks and private versions watermarked with different keys.

G.2 MEMBERSHIP INFERENCE

Membership inference, initially proposed by Shokri et al. (2017), is a long-standing problem in machine learning: given a data point and a machine learning model, determine whether that data point was used to train the model. MIAs for LLMs are broadly based on applying pre-defined thresholds to membership scores that are typically based on *loss-based* metrics. We briefly describe the specific membership scores proposed by different MIAs that we employ in our experiments.

- PERPLEXITY: Proposed by Yeom et al. (2018), this MIA uses loss (perplexity in the context of LMs) as the scoring metric. However, this approach suffers from high false positives as it tends to classify naturally predictable sequences as members of the training set.
- ZLIB ENTROPY (Carlini et al., 2021) computes a score by taking the ratio between the model's perplexity and the zlib compression size of the text. Lower ratios indicate potential membership in the training data.
- MIN-K% Shi et al. (2024) computes the score by averaging the probabilities of the $k\%$ least likely tokens in a sequence. By focusing on the least likely tokens, it aims to solve the false positive problem with perplexity.
- MIN-K%++ Zhang et al. (2024b) compares the probability of the target token with the expected probability of all tokens within the vocabulary. It is based on the insight that each training token will tend to have higher probability relative to many other candidate tokens in the vocabulary.
- DC-PPD Zhang et al. (2024c) computes the divergence between the token probability distribution and the token frequency distribution for detection.

G.3 DETECTING DATASET MEMBERSHIP

Detecting dataset membership addresses the challenge of detecting whether a given dataset was used by LLM developers in pretraining. Unlike membership inference attacks (MIAs), which focus on identifying whether individual sequences were included in a model's training data, dataset membership concerns verifying the inclusion of a collection of documents.

Wei et al. (2024) propose a hypothesis-testing approach to detect membership by inserting random sequences or Unicode character substitutions as data watermarks. This method works by testing the model's preference for the inserted data watermarks against other random data watermarks. First, their proposed watermarks can impact machine readability, affecting search engine indexing and retrieval-augmented generation (RAG) pipelines. More critically, unicode substitutions can significantly alter tokenization processes, potentially compromising the utility of evaluation benchmarks. Although these limitations may be manageable for some creators, Our approach offers an alternative that better preserves content quality while maintaining detection capability. Another recent proposal Maini et al. (2024) is to selectively combining MIAs that provide positive signal for a given distribution, and aggregating them to perform a statistical test on a given dataset. Their method assumes access to a *validation* set drawn from the same distribution as the target dataset and unseen by the model–a requirement that can be challenging to satisfy in many practical scenarios.

Meeus et al. (2024a) propose inserting "copyright traps" into documents to enhance document-level membership inference for smaller models that lack natural memorization. Liu et al. (2023b) introduce a backdoor-based dataset inference approach. However, these methods rely on heuristics and do not provide the false positive guarantees that hypothesis-testing-based approaches offer.

Recent studies Maini et al. (2024); Duan et al. (2024); Das et al. (2024); Meeus et al. (2024b) suggest that detecting sequence level membership in LLMs trained on trillions of tokens in a single epoch is likely infeasible. These studies also highlight the limited efficacy of MIAs for LLMs, showing that such approaches barely outperform random guessing. Moreover, the apparent success of MIAs in certain scenarios can often be attributed to distributional differences between the *member* and *non-member* sets used in evaluations, rather than their ability to reliably infer true membership.

G.4 Test Set Contamination Detection

There have been a few recent third-party approaches that are focused on detecting test-set contamination in LLMs. Heuristic prompting-based methods Sainz et al. (2023b); Golchin & Surdeanu (2024) attempt to detect contamination by prompting models to reproduce exact or near-exact test examples. Reproducing verbatim examples requires a high level of memorization which typically requires a high duplication of test examples Carlini et al. (2021) and strong memorization capabilities typically absent in smaller models Meeus et al. (2024a). The heuristic nature of these approaches prevents them from providing a statistical evidence of contamination.

Statistical approaches to detect contamination are limited. Oren et al. (2023) build on the principle that in absence of data contamination, all orderings of an *exchangeable* test set should be equally likely. Their work relies on the strong assumption of metadata contamination (canonical ordering of the dataset)–a presumption that can often be violated. Another recent proposal Zhang et al. (2024a) uses a statistical test to compare model confidence on original test instances and their rephrased counterparts. However, as discussed earlier, their null hypothesis can be invalid due to LLMs' inherent bias towards machine-generated content.

H Case Study: Detecting Research Paper Abstracts in Pretraining Data

To demonstrate the broader applicability of **STAMP** for detecting dataset membership across different forms of content, we explore its effectiveness in detecting membership of abstracts of papers from EMNLP '24 proceedings Al-Onaizan et al. (2024). We evaluate both the preservation of academic writing quality in watermarked abstracts and the effectiveness of **STAMP** in detecting their inclusion in training data.

Experimental Setup. We sample 500 papers from EMNLP 2024 proceedings and create watermarked versions of their abstracts following our methodology from Section 3. The prompt template and examples of rephrased abstracts are presented in Appendix J and Appendix K.2 respectively. To evaluate detection capability, we perform controlled experiments on the Pythia 1B model (Biderman et al., 2023) through continual pretraining. The pretraining corpora consists of a mixture of the public watermarked versions of these abstracts and a subset of OpenWebText (approximately 3 billion tokens). The abstracts comprise approximately 100K tokens, representing just 0.003% of the pretraining corpus.

Results. Table 10 demonstrates **STAMP**'s effectiveness in detecting dataset membership. Our approach achieves a near-zero p-value ($\approx 10^{-12}$), indicating strong statistical evidence of membership. For comparison, LLM DI Maini et al. (2024) achieves a p-value of 0.05 with access to a *validation* set of unseen abstracts from the same conference and is unable to detect membership using the privately held counterparts of the same abstracts included in the pretraining data as the *validation* set. In Table 8 we evaluate state-of-the-art MIAs and finding that they perform no better than random chance (AUROC ≈ 0.5). Our findings corroborate with recent studies Duan et al. (2024); Maini et al. (2024); Das et al. (2024) that highlight the failure of sequence level MIAs on LLMs.

Quality Evaluation. To evaluate the quality of watermarked abstracts, we use GPT-4 OpenAI (2024) as a judge following the prompt template in Figure H. Each abstract was classified into one of five quality tiers. Our analysis shows that 82.7% of the watermarked abstracts were rated as *preferred* and 16.3% as *acceptable* indicating that 99% maintain high academic quality. Only 1% required *minor revisions*, with none requiring *major revisions* or deemed *inadequate*.

Since LLMs often exhibit systematic preferences for LLM-generated text over human-written content (Liu et al., 2023a; Mishra et al., 2023; Laurito et al., 2024), we additionally conduct a human study involving the original authors. We asked 40 authors to rate watermarked versions of their own abstracts using the same quality tiers. The human evaluation strongly corroborates our automatic assessment, with most watermarked versions being *preferred* or *acceptable*: 4 authors *preferred* the watermarked version, 24 authors rated the watermarked abstracts as *acceptable*, 11 indicated the text required *minor revisions* and just 1 indicating that their rephrased abstract requires major edits.

Additionally, we measure semantic preservation using the P-SP metric Wieting et al. (2021), finding an average score of **0.95** between original and watermarked abstracts, demonstrating strong semantic similarity.

Table 10: **Comparison of different approaches for detecting membership of paper abstracts.**

METHOD	P-VALUE
LLM DI MAINI ET AL. (2024) (1)	0.15
LLM DI MAINI ET AL. (2024) (2)	0.05
STAMP (W/O PAIRED TESTS)	0.01
STAMP	2.7E-12

Prompt Template to Evaluate Quality of the Rephrased Abstracts using GPT4

```
You will be given an original abstract and its rephrased
version.  Your task is to evaluate the quality of abstract
rewrites for ML research paper based on:

1.  Meaning Preservation
2.  Clarity
3.  Technical Accuracy

Evaluate the rewritten abstract and assign one of these
ratings:
- Preferred:  The rewrite improves upon the original in terms
of clarity and readability while maintaining full technical
accuracy.
- Acceptable:  The rewrite matches the original in quality
and could serve as a direct replacement without requiring
changes.
- Minor Revisions:  The rewrite is promising but requires
minor edits to reach the original's quality.
- Major Revisions:  The rewrite has significant issues with
meaning preservation, clarity, or technical accuracy and
requires major edits.
- Inadequate:  The rewrite fails to convey the original
research effectively due to critical flaws in meaning,
clarity, or technical accuracy.

Here are the abstracts:

Original Abstract:  {original_abstract}
Rephrased Abstract:  {watermarked_abstract}

Provide a short explanation of your rating, followed by your
final rating in the format:
Final Rating:  {rating}
```

I CASE STUDY: DETECTING ML BLOG POSTS IN PRETRAINING DATA

The inclusion of copyrighted material in LLM training data has emerged as a significant concern, leading to legal disputes, such as the lawsuit between New York Times and OpenAI NewYorkTimes (2023), among others. Through a case study, we demonstrate how **STAMP** can help creators detect potential unauthorized use of their content in model training. Specifically, we use **STAMP** to detect the membership of the popular AI Snake Oil newsletter Narayanan & Kapoor (2023).

Experimental Setup. We collect 56 blogs from the newsletter, creating watermarked versions of each newsletter using, the prompt template is presented in Figure J. We randomly select a subset of 44 blogs that we include in pretraining corpora and keep the remaining 12 blogs as a *validation* set that is unseen by the model. To evaluate detection capability, we perform controlled experiments on the Pythia 1B model (Biderman et al., 2023) through continual pretraining. The pretraining corpora consists of a mixture of the public watermarked versions of these abstracts and a subset of OpenWebText (approximately 3 billion tokens). The abstracts comprise approximately 94K tokens, representing just 0.003% of the pretraining corpus.

Table 11: **Comparison of different approaches for detecting membership of AI Snake Oil.**

METHOD	P-VALUE
LLM DI MAINI ET AL. (2024) (1)	0.44
LLM DI MAINI ET AL. (2024) (2)	0.58
STAMP (W/O PAIRED TESTS)	0.07
STAMP	2.4E-3

Results. Table 10 demonstrates **STAMP**'s effectiveness in detecting dataset membership for the blog articles. LLM DI is unable to detect membership under the two different choices of validation set: (1) with the private rephrases of the same 44 blog posts as the validation set, and (2) with the version of the *held out* set of 12 blog posts that is watermarking using the public key. In Table 8 we evaluate state-of-the-art MIAs and finding that they perform no better than random chance (AUROC ≈ 0.5). Our findings corroborate with recent studies Duan et al. (2024); Maini et al. (2024); Das et al. (2024) that highlight the failure of sequence level MIAs on LLMs.

J PROMPT TEMPLATES FOR REPHRASING

In this section, we outline the prompts used with LLaMA-3 70B AI@Meta (2024) to generate watermarked versions of the documents used in our experiments.

Prompt Template for Rephrasing Benchmarks

```
Rephrase the question given below.  Ensure you keep all
details present in the original, without omitting anything
or adding any extra information not present in the original
question.

Question:  What is the main energy source for deep ocean
currents that move large volumes of water around the planet?

Your response should end with "Rephrased Question:
[rephrased question]"
```

Prompt Template for Rephrasing Abstracts

```
Rephrase the abstract of a ML research paper given below
following these strict guidelines:

PRESERVE:
- All technical details and findings
- Original tone of the abstract

AVOID:
- Adding interpretive language not present in the original
abstract
- Removing any details
- Changing meaning or emphasis

Abstract: {original_abstract}

Your response should end with "Rephrased Abstract:
{rephrased_abstract}"
```

Prompt Template for Rephrasing Blogs

```
Rephrase the below paragraph from an AI newsletter while
maintaining coherent flow between paragraphs. Here are your
instructions:

1. I will provide the previous paragraph (marked as CONTEXT)
and the current paragraph to rephrase (marked as TARGET).
2.Your task is to:
- Rephrase the TARGET paragraph so it flows naturally from
the previous paragraph (CONTEXT)
- Keep the same tone and emphasis as the original paragraph
-Preserve the technical details present in the original
paragraph
- Do not add any extra information not present in the
original paragraph
- Avoid making sentences wordier or adding interpretive
language

3. Format your response as: REPHRASED PARAGRAPH: [your
rephrased version]

Context: {context}
Paragraph: {paragraph}
```

K WATERMARKED EXAMPLES

K.1 WATERMARKED TEST SETS

K.1.1 TRIVIAQA

Original Question: Which enduring cartoon character was created by Bob Clampett for the 1938 cartoon Porky's Hare Hunt?

Rephrased Question: Which long-lasting cartoon character was originally created by Bob Clampett for the 1938 cartoon titled 'Porky's Hare Hunt'?

Original Question: Which US state lends its name to a baked pudding, made with ice cream, sponge and meringue?

Rephrased Question: Which US state is the namesake of a baked pudding that consists of sponge, meringue, and ice cream?

K.1.2 ARC CHALLENGE

Original Question: Company X makes 100 custom buses each year. Company Y makes 10,000 of one type of bus each year. Which of the following is the most likely reason a customer would buy a bus from company X instead of company Y?

Rephrased Question: What is the most probable reason a customer would choose to purchase a bus from Company X, which produces 100 custom buses annually, over Company Y, which manufactures 10,000 buses of a single type each year?

Original Question: Sugars are necessary for human cell function. Which of the following are human cells not capable of doing?

Rephrased Question: Given that sugars are necessary for human cell function, what is it that human cells are unable to do?

K.1.3 MMLU

Original Question: Noradrenaline is the neurotransmitter between which of the two structures below?

Rephrased Question: Between which two structures listed below does noradrenaline act as the neurotransmitter?

Original Question: On which surfaces of the teeth is dental plaque most likely to accumulate in the mouth of a patient with poor oral hygiene?

Rephrased Question: In a patient with poor oral hygiene, on which surfaces of the teeth is dental plaque accumulation most probable in the mouth?

K.1.4 GSM8K

Original Question: Darrell and Allen's ages are in the ratio of 7:11. If their total age now is 162, calculate Allen's age 10 years from now.

Rephrased Question: If the current ages of Darrell and Allen are in a 7:11 ratio and their combined age at present is 162, what will be Allen's age a decade from now.

Original Question: Indras has 6 letters in her name. Her sister's name has 4 more letters than half of the letters in Indras' name. How many letters are in Indras and her sister's names?

Rephrased Question: If Indras' name contains 6 letters, and her sister's name has a number of letters that is 4 more than half the number of letters in Indras' name, then how many letters are there in Indras' and her sister's names combined?

K.2 WATERMARKED PAPER ABSTRACTS

Tree of Problems: Improving structured problem solving with compositionality Zebaze et al. (2024)

Original Abstract:

Large Language Models (LLMs) have demonstrated remarkable performance across multipletasks through in-context learning. For complex reasoning tasks that require step-by-step thinking, Chain-of-Thought (CoT) prompting has given impressive results, especially when combined with self-consistency. Nonetheless, some tasks remain particularly difficult for LLMs to solve. Tree of Thoughts (ToT) and Graph of Thoughts (GoT) emerged as alternatives, dividing the complex problem into paths of subproblems. In this paper, we propose Tree of Problems (ToP), a simpler version of ToT, which we hypothesise can work better for complex tasks that can be divided into identical subtasks. Our empirical results show that our approach outperforms ToT and GoT, and in addition per forms better than CoT on complex reasoning tasks. All code for this paper will be made available.

Watermarked Abstract:

Large Language Models (LLMs) have achieved notable performance in multiple tasks using in-context learning. For complex reasoning tasks requiring step-by-step thinking, Chain-of-Thought (CoT) prompting, especially when combined with self-consistency, has yielded impressive results. However, certain tasks continue to pose significant challenges for LLMs. Tree of Thoughts (ToT) and Graph of Thoughts (GoT) have emerged as alternatives by breaking down complex problems into subproblem paths. This paper introduces Tree of Problems (ToP), a simplified ToT version that we hypothesize will be more effective for complex tasks comprising identical subtasks. Our empirical findings indicate that ToP outperforms both ToT and GoT, and also surpasses CoT on complex reasoning tasks. The code used in this paper will be made available.

"A good pun is its own reword": Can Large Language Models Understand Puns? Xu et al. (2024)

Original Abstract:

Puns play a vital role in academic research due to their distinct structure and clear definition, which aid in the comprehensive analysis of linguistic humor. However, the understanding of puns in large language models (LLMs) has not been thoroughly examined, limiting their use in creative writing and humor creation. In this paper, we leverage three popular tasks, i.e., pun recognition, explanation and generation to systematically evaluate the capabilities of LLMs in pun understanding. In addition to adopting the automated evaluation metrics from prior research, we introduce new evaluation methods and metrics that are better suited to the in-context learning paradigm of LLMs. These new metrics offer a more rigorous assessment of an LLM's ability to understand puns and align more closely with human cognition than previous metrics. Our findings reveal the \lazy pun generation" pattern and identify the primary challenges LLMs encounter in understanding puns.

Watermarked Abstract:

Puns, due to their distinct structure and clear definition, play a vital role in academic research on linguistic humor analysis. Despite their importance, the understanding of puns in large language models (LLMs) remains understudied, which hinders their application in creative writing and humor generation. This paper employs three tasks - pun recognition, explanation, and generation - to systematically evaluate LLMs' capabilities in pun understanding. Building upon prior research's automated evaluation metrics, we propose new evaluation methods and metrics tailored to the in-context learning paradigm of LLMs. These novel metrics provide a more rigorous assessment of an LLM's pun understanding ability and better align with human cognition. Our results expose the "lazy pun generation" pattern and pinpoint the primary challenges LLMs face in understanding puns.

MiniCheck: Efficient Fact-Checking of LLMs on Grounding Documents Tang et al. (2024)

Original Abstract:

Recognizing if LLM output can be grounded in evidence is central to many tasks in NLP: retrieval-augmented generation, summarization, document-grounded dialogue, and more. Current approaches to this kind of fact-checking are based on verifying each piece of a model generation against potential evidence using an LLM. However, this process can be very computationally expensive, requiring many calls to a model to check a single response. In this work, we show how to build small fact-checking models that have GPT-4-level performance but for 400x lower cost. We do this by constructing synthetic training data with GPT-4, which involves creating realistic yet challenging instances of factual errors via a structured generation procedure. Training on this data teaches models to check each fact in the claim and recognize synthesis of information across sentences. For evaluation, we unify datasets from recent work on fact-checking and grounding LLM generations into a new benchmark, LLM-AggreFact. Our best system MiniCheck-FT5 (770M parameters) outperforms all systems of comparable size and reaches GPT-4 accuracy. We release LLM-AggreFact, code for data synthesis, and models.

Watermarked Abstract:

Verifying the evidential grounding of LLM outputs is crucial for various NLP tasks, including retrieval-augmented generation, summarization, and document-grounded dialogue. Existing fact-checking approaches typically involve verifying each component of model generation against potential evidence using an LLM, which can be computationally expensive and require multiple model calls for a single response. This work demonstrates the construction of compact fact-checking models that achieve GPT-4-level performance at a 400x lower cost. We accomplish this by generating synthetic training data using GPT-4 through a structured procedure that creates realistic yet challenging instances of factual errors. Models trained on this data learn to verify facts within claims and recognize information synthesis across sentences. We establish a unified benchmark, LLM-AggreFact, by consolidating datasets from recent fact-checking and LLM grounding research. Our top-performing system, MiniCheck-FT5 (770M parameters), outperforms comparable-sized systems and matches GPT-4's accuracy. We make LLM-AggreFact, the data synthesis code, and the models publicly available.