Perturb Your Data: Paraphrase-Guided Training Data Watermarking

Pranav Shetty, Mirazul Haque, Petr Babkin, Zhiqiang Ma, Xiaomo Liu, Manuela Veloso
JPMorgan AI Research
{first.last}@jpmchase.com

Abstract

Training data detection is critical for enforcing copyright and data licensing, as Large Language Models (LLM) are trained on massive text corpora scraped from the internet. We present SPECTRA, a watermarking approach that makes training data reliably detectable even when it comprises less than 0.001% of the training corpus. SPECTRA works by paraphrasing text using an LLM and assigning a score based on how likely each paraphrase is, according to a separate scoring model. A paraphrase is chosen so that its score closely matches that of the original text, to avoid introducing any distribution shifts. To test whether a suspect model has been trained on the watermarked data, we compare its token probabilities against those of the scoring model. We demonstrate that SPECTRA achieves a consistent p-value gap of over nine orders of magnitude when detecting data used for training versus data not used for training, which is greater than all baselines tested. SPECTRA equips data owners with a scalable, deploy-before-release watermark that survives even large-scale LLM training.

1 Introduction

Contemporary large language models (LLMs) utilize extensive datasets sourced from the internet for pretraining, which lead to their general-purpose abilities, but may include content scraped without permission. Although these large corpora are necessary for the emergent abilities of LLMs, this practice raises several ethical and legal concerns. The utilization of copyrighted material in the training process may violate its licensing terms. Many open-weight models are released without disclosing their training data, leaving model end-users vulnerable to liability for damages and hindering the adoption of open-source models.

Several recent lawsuits have focused on the unauthorized use of pay-walled data for training models [33, 5]. As more content is consumed online through various intermediate LLMs like ChatGPT, it becomes vital to ensure that content creators are appropriately incentivized to produce new content. Otherwise, the data collection practices of most LLM providers may lead to an "extractive dead end" [26]. Consequently, there is an important need to detect unauthorized use of data for training.

Many recent techniques have been developed to detect training data [39–41]. Membership Inference Attack (MIA) techniques build on the idea that training leads to changes in the log probabilities output by the model, which is measured by comparing these log probabilities (MIA scores) for training data (also called member data) against a held-out or non-member dataset from the same domain that was not used for training. A content creator must provide the suspect data and held-out data, which can then be tested on a target model. However, MIA methods are sensitive to small distributional shifts between the suspect and held-out data, which can lead to spurious performance [8, 19]. Furthermore,

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: The First Workshop on Generative and Protective AI for Content Creation.

since Membership Inference Attack (MIA) techniques inherently require held-out data for validation, content creators lacking access to such data are unable to perform validation.

Given the limitations of standard MIA techniques, it is necessary to find alternative methods for content creators to protect their content. STAMP [25] is a recent effort in this direction that watermarks text using the KGW scheme [14] multiple times with different keys, with one version made public and the rest kept private by the content creator. If their data is used for training, content creators can compare the perplexity of the public rephrases against the private rephrases to establish membership (included in the training data). In contrast to MIA, which detects the membership of each document, STAMP performs inference over the entire dataset, enabling it to be robust to noise in individual examples. STAMP requires storing a large number of private rephrases for each document. This method also requires modifying the decoding layer of an LLM to generate watermarked text, which may be inaccessible due to the large amount of GPU resources required. The p-values between true positives and false positives are at most three orders of magnitude apart in STAMP, which we argue is not sufficient for such a task with significant legal implications.

To address these limitations, we introduce **S**core sampled re**P**hrasing to det**EC**t **TRA**ining data (SPECTRA) (Figure 1). We assume that a content creator creates textual content (and holds the copyright) and, before making it publicly available, watermarks a portion of it using SPECTRA. Such content might be openly accessible, placed behind a paywall, or protected by licenses explicitly prohibiting web scraping for model training. If the content creator suspects a particular model of unauthorized use of their data, they or an authorized intermediary can conduct a statistical test to confirm whether the watermarked data was indeed part of that model's training set.

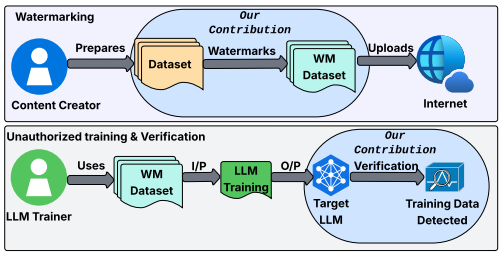

Figure 1: Overview of our problem setting. I/P and O/P refer to input and output, respectively.

The SPECTRA approach can be divided into two phases: watermarking and verification. During *watermarking*, SPECTRA generates several paraphrases from an LLM without needing to modify the decoding process. A score that provides a membership signal is computed over the paraphrases using a scoring model that has not been trained on them. Specifically, we compute the Min-K%++ [40] score, which is the normalized log probability averaged over the lowest $K\%$ tokens (highest surprisal tokens) in a document. Our key insight is that existing MIA scores are designed to measure changes in the loss surface between a model before and after it is trained on specific data, but in practice, we seldom have access to the model before training. We employ our scoring model as a proxy for this initial (pre-trained) state, allowing us to detect changes attributable to training reliably. To avoid introducing systematic bias, paraphrases are carefully selected using a sampling strategy that we designed, such that their Min-K%++ scores remain close to the original document's score. This constraint ensures that the watermarking procedure itself does not introduce any false-positive membership signals. In the *verification* phase, we construct a statistical test in which the scores from the Min-K%++ scores from the scoring model are compared against the scores from the suspected model to obtain a p-value. Unlike other MIA methods, SPECTRA does not require a non-member dataset. We find that SPECTRA yields a statistically significant result for identifying membership in all datasets used for training, without yielding false positives. SPECTRA will equip content creators with the tools to enforce their intended data-use policies.

Our contributions are as follows:

1. We show that SPECTRA can be used to watermark pre-training data and that the watermark can be measured after continued pre-training with 5 billion tokens. Each of our datasets constituted less than 0.001% of the training data.

2. We design a strategy to sample paraphrases that reduces false positives and outperforms other sampling approaches, such as random sampling or selecting paraphrases with the maximum Min-K%++ score.

3. We benchmark against existing methods for detecting training data and find that SPECTRA is the only one that yields a statistically significant result to identify membership for all datasets tested. Moreover, SPECTRA gave the largest change in p-values between false positives and true positives, yielding a consistent difference of at least nine orders of magnitude across datasets.

2 Related Work

Membership Inference Attack. In the context of LLMs, recent studies have proposed several Membership Inference Attack (MIA) methods that use scores derived from the log probabilities output by the model to differentiate member data (used during training) from non-member data (not used during training). The datasets employed to evaluate these methods, such as WikiMIA [29] and PatentMIA [41], were constructed by collecting data published before (member) and after (non-member) the LLM's training cutoff date. It was later observed that these MIA techniques primarily detected temporal artifacts. Subsequent work showed that when member and non-member datasets were sampled homogeneously, all tested MIA methods performed no better than random classifiers [8, 19, 7].

Data Watermarks. Data watermarks are modifications made to a text to make it more detectable when used to train a model. Wei et al. [34] insert hash strings into a model and show that models trained on such hashes occurring multiple times in the dataset memorize the hash. Other works have proposed backdoor attacks that insert carefully picked tokens into the text, which can be detected post-training by using a secret prompt [4, 38]. The assumption underlying these works is that LLM trainers collect large quantities of data from the internet and are likely to collect data with such watermarks. The challenge is that these affect the meaning and readability of the text and are thus not suitable when the text is meant for human readers.

Dataset Inference. In contrast to MIA, Maini et al. [19] proposes Dataset Inference (DI), where the goal is not to obtain accurate labels over every document in the dataset but to obtain a measure of confidence over the entire dataset being tested. As LLM trainers tend to scrape each source comprehensively, it is likely that related documents from a single source will all be used for training. Using multiple documents also enhances the signal available for detecting membership and makes the inference less susceptible to noise due to outliers. Our work takes inspiration from this.

3 Problem Setup

Consider a content creator who possesses a dataset D, which they wish to make available online. D could consist of articles from sources such as news providers or academic publishers. The creator, however, wishes to prevent unauthorized use of this dataset for training LLMs.

To enable detection of unauthorized use, the creator applies a watermarking procedure W to transform the dataset D into a modified dataset $D' = W(D)$. The creator retains D and only publishes D'.

After publication, the content creator may seek to test whether a particular model M has been trained on the watermarked dataset D'. We assume a grey-box setting, where the model M is queried and provides log probabilities over tokens, but the model weights and architecture are not necessarily known. This scenario is typical of open-source models, some of which are used commercially. For closed-source models, testing may be facilitated through a neutral third-party arbiter with grey-box access (say, a court-appointed arbiter). Given D, D' and grey-box access to M, the content creator or arbiter applies a statistical test T to determine if the model M was trained on D'. Note that this procedure detects membership of the entire dataset and not each document, which can be noisier.

The research question we address is: How can we optimally design the watermarking procedure W and the statistical test T such that:

1. If the model M is indeed trained on the watermarked dataset D', then T reliably identifies the true positives with high confidence.

2. Conversely, if M is *not* trained on D', then T does not yield false-positive outcomes.

3.1 Training Data Detection Signals

Some common scores in the literature that are computed to determine membership in the training data of a model are described below. In our scenario, M is an autoregressive language model that generates a probability distribution for each subsequent token, denoted as $P(x_t \mid x_{<t}; M)$.

1. **Loss** [39]: This method relies on measuring the loss of a given target sequence x under the model M. The membership inference score is directly defined as: $f(x; M) = L(x; M)$ where $L(x; M)$ denotes the negative log-likelihood (loss) of the target sequence according to the model M.
2. **Min-K%** [29]: This method focuses specifically on the $K\%$ of tokens that have the lowest likelihood under the model M. The membership inference score is computed as the average log probability over these tokens.
3. **Min-K%++** [40] Min-K%++ enhances the original Min-K% score by incorporating normalization relative to the mean and variance of token log probabilities.

Other MIA scores are discussed in Appendix E. We focus on Min-K%++ in this work, which we describe in greater detail next.

3.2 Min-K%++ score

Formally, given an autoregressive model M and a token sequence $x = (x_1, x_2, \ldots, x_n)$, define the token-level normalized log probability as $z(x_t; M) = \frac{\log P(x_t|x_{<t};M) - \mu_{x_{<t}}}{\sigma_{x_{<t}}}$ where

$$\mu_{x_{<t}} = \mathbb{E}_{z\sim P(\cdot|x_{<t};M)}[\log P(z \mid x_{<t}; M)],$$
$$\sigma_{x_{<t}} = \sqrt{\mathbb{E}_{z\sim P(\cdot|x_{<t};M)}\left[(\log P(z \mid x_{<t}; M) - \mu_{x_{<t}})^2\right]}.$$

Here, $\mu_{x_{<t}}$ represents the expectation of the log probability distribution for the next token given the prefix $x_{<t}$, and $\sigma_{x_{<t}}$ denotes the corresponding standard deviation.

The Min-k%++ score for a sequence x is defined as the average of the normalized log probabilities $z(x_t; M)$ over the k% of tokens in the sequence with the lowest values (indicating highest surprisal):

$$f_{\text{Min-k\%++}}(x; M) = \frac{1}{|\text{min-k}(x)|} \sum_{x_t \in \text{min-k}(x)} z(x_t; M).$$

This normalization allows Min-K%++ to better distinguish sequences that are part of the training data from those that are not, by highlighting the relative surprisal of the most unlikely tokens while making it robust to absolute probability shifts across tokens. Critically, prior work [40] has shown that the Min-K%++ score theoretically corresponds to measuring the negative trace of the Hessian matrix of the log-likelihood $log\ P(x_t \mid x_{<t})$. Intuitively, training via maximum-likelihood directly reduces the curvature (Hessian trace) of the loss landscape at training examples, thereby causing their corresponding Min-K%++ scores to increase. We evaluate the performance of various MIA scores on three datasets (Table 1) that were not used during the pretraining of Pythia models. When the datasets, each containing 500 samples (at most 512 tokens each), are used for training with an additional 500 million text tokens, Min-K%++ has the best performance among all methods. However, as reported by prior work [8], the effectiveness of Min-K%++ diminishes significantly at larger scales of training (5 billion tokens), dropping to performance indistinguishable from random. Based on these observations, we hypothesize that while Min-K%++ inherently captures strong training signals, its effectiveness at large scales may be hindered by distributional homogeneity when measured against non-member data, and that changing the reference for measurement would enable us to capture a stronger training signal.

4 Watermarking with SPECTRA

This phase takes place after a content creator writes their content and before they publish it. To watermark a dataset, we generate multiple paraphrases of each document using an LLM and compute their Min-K%++ scores using a separate scoring model that has not been trained on the dataset being

Table 1: ROC-AUC of classifying training data used for continued pretraining of a Pythia 410m model against a held-out dataset from the same domain. The ROC-AUC is computed using the MIA scores below. 500 million or 5 billion tokens are used during continued pretraining.

	Metrics	Wiki	HN	PubMed
Datasets +500 million tokens	Loss	0.71	0.73	0.63
	DC-PDD	0.77	0.79	0.64
	Min-k %	0.76	0.79	0.65
	Min-K%++	**0.85**	**0.84**	**0.72**
Datasets +5 billion tokens	Loss	0.55	0.54	0.52
	DC-PDD	0.55	0.52	0.50
	Min-k %	0.56	0.55	0.52
	Min-K%++	0.55	0.55	0.51

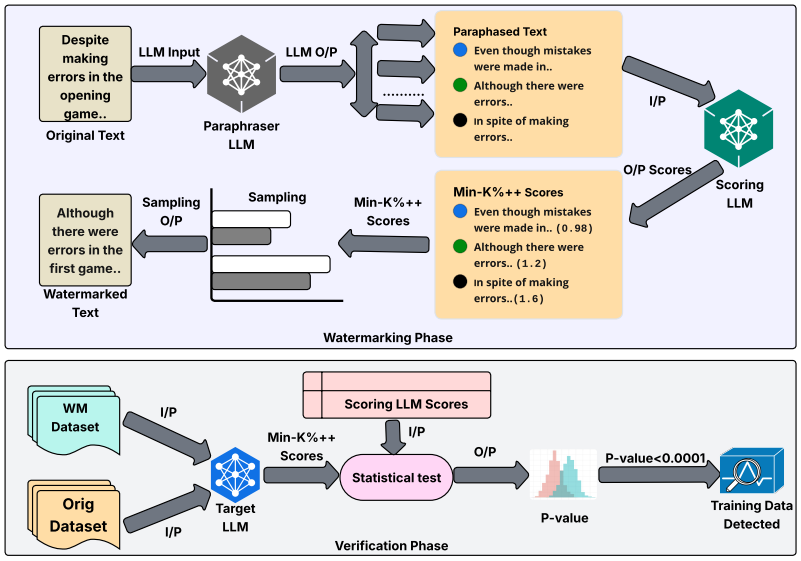

Figure 2: Overview of SPECTRA. **Watermarking Phase**: We use an LLM to generate multiple paraphrases of the original text. We sample one paraphrase that has a Min-K%++ score close to the original text. **Verification Phase**: Given a target LLM suspected of being trained on the watermarked data, we compute the Min-K%++ scores of the watermarked and original data and compare against the scores previously generated by the scoring model. Membership is detected through a paired t-test.

watermarked. In practice, such a model is easy to find as D and D' are unpublished at the time of scoring. We sample one paraphrase as the watermarked sample according to Algorithm 1.

The sampling favors paraphrases with scores close to the original score. For each side (above or below the original), we define a categorical distribution over candidate indices using weights proportional to $exp(-\alpha|r_{ij} - 1|)$ where $\alpha > 0$ controls the sharpness of the distribution. A higher α causes the algorithm to favor paraphrases with scores close to the original more strongly. In practice, we pick the largest α that does not lead to numerical underflow issues ($\alpha = 100$).

If both above- and below-original paraphrases are available, the side is chosen probabilistically in inverse proportion to how often each type appears globally across the dataset, helping to avoid systematic score shifts. This sampling strategy ensures that the distribution of scores for the watermarked dataset remains similar to the original, reducing the likelihood of false positives when testing against models not trained on D'. Importantly, because the Min-K%++ scores of watermarked text tend to increase after training, the score distribution after training becomes distinguishable from the pre-training state. Notably, if paraphrases were consistently selected only from the high-score side, it

Algorithm 1 Sampling paraphrases

Input: Original scores $\{s_i^{(0)}\}_{i=1}^N$, paraphrased scores $\{S_i = \{s_{i1}, \ldots, s_{im}\}\}_{i=1}^N$, paraphrases $\{T_i = \{t_{i1}, \ldots, t_{im}\}\}_{i=1}^N$, parameter $\alpha = 100$
Output: Sampled paraphrases $\{t_{ij_i}\}_{i=1}^N$

1. Pre-computation: define $r_{ij} = s_{ij}/s_i^{(0)}$, then $\mathcal{A}=\{i : r_{ij} < 1, \forall j\}$, $\mathcal{B}=\{i : r_{ij} > 1, \forall j\}$.
2. Global side-balance: $\pi_+ = \begin{cases} 0.5 & \text{if } |\mathcal{A}| + |\mathcal{B}| = 0 \\ \frac{|\mathcal{A}|}{|\mathcal{A}|+|\mathcal{B}|} & \text{otherwise} \end{cases}$, $\quad \pi_- = 1 - \pi_+$.
3. For each datapoint $i = 1, \ldots, N$:
   (a) Partition: $\mathcal{R}_i^{(-)} = \{j : r_{ij} < 1\}$, $\mathcal{R}_i^{(+)} = \{j : r_{ij} > 1\}$.
   (b) Side s_i: if one set empty, choose the other; else sample from $\{+, -\}$ w.p. π_+, π_-.
   (c) Let $\mathcal{R} = \mathcal{R}_i^{(s_i)}$.
   (d) Weights: $w_{ij} = \exp(-\alpha|r_{ij} - 1|)$ for $j \in \mathcal{R}$.
   (e) Normalize: $w_{ij} \leftarrow w_{ij} / \sum_{k\in\mathcal{R}} w_{ik}$.
   (f) Sample j_i from categorical$\{w_{ij}\}_{j\in\mathcal{R}}$.
4. Return $\{t_{ij_i}\}_{i=1}^N$.

would create a detectable signature even without training—leading to false positives. SPECTRA avoids this by balancing selection, ensuring a reliable signal only when training has occurred.

4.1 Verification of training with SPECTRA

During this phase, the content is released to the public and it is suspected that the data may have been used in an unauthorized manner for training. Given an original datapoint $x \in D$ and its watermarked counterpart $x' \in D'$, we define the score ratio under a model M as $r(x, x'; M) = \frac{f_{\text{Min-k\%++}}(x';M)}{f_{\text{Min-k\%++}}(x;M)}$. Let M_S denote the scoring model (not trained on D'), M_T denote the target model (potentially trained on D'), and M_U denote the checkpoint of M_T before it was trained on D'. Given that Min-K%++ scores rise after training, we can write $\mathbb{E}_{x'\in D'}[f_{\text{Min-k\%++}}(x'; M_T)] > \mathbb{E}_{x'\in D'}[f_{\text{Min-k\%++}}(x'; M_U)]$. However, in practice, we do not have access to M_U and so we approximate it using M_S. However, because M_S and M_T may differ in architecture or baseline predictions, we normalize each watermarked score by the corresponding original document score to allow meaningful comparisons between the two models. Thus, under the null hypothesis H_0, the ratio of scores under M_T is equal to that under M_S when D' is not used for training M_T, i.e., $H_0 : \mathbb{E}_{x\in D, x'\in D'}[r(x, x'; M_T)] = \mathbb{E}_{x\in D, x'\in D'}[r(x, x'; M_S)]$. The alternate hypothesis H_1 states that the ratio of scores under the target model is lower relative to the scoring model when D' was used for training, i.e., $H_1 : \mathbb{E}_{x\in D, x'\in D'}[r(x, x'; M_T)] < \mathbb{E}_{x\in D, x'\in D'}[r(x, x'; M_S)]$. Note that the inequality flips sign here as Min-K%++ values in the denominator are always negative. We test these hypotheses by computing these ratios for all pairs (x, x') in each dataset and performing a 1-sided paired t-test. A low p-value would indicate rejection of the null hypothesis H_0, providing statistically significant evidence that the target model M_T has indeed been trained on the watermarked dataset D'.

5 Results

Datasets: We use datasets for training that have not previously been used to train our target model of interest. Thus, we are limited to training models where the pre-training data is known transparently. Two prominent models that meet this criterion are the Pythia models and the OLMo [11] models, with the corresponding training datasets being The Pile and Dolma.

1. **The Pile**: We use the deduplicated subsets of the Pile [9, 8] from the domains of Wikipedia (Wiki), Hackernews (HN), and Pubmed Central abstracts (PubMed) that were held out from training.

2. **Dolma**: We use the PeS2o held-out subset of Dolma obtained from Paloma [31, 18]. All text in this subset was released after the release date of the Pile, making it non-member for Pythia models.

   See Appendix A for additional details.

Models: The watermarking pipeline consists of 3 different types of models:

1. **Paraphraser model**: We use the LLama-3.1-405b model and generate 10 paraphrases per document [10] (Appendix B).
2. **Scoring model**: We use the Pythia 2.8b-deduped model for the Pile datasets. We use a model that is known not to have been trained on our datasets, as otherwise, the distribution of the Min-K%++ scores would shift higher. Pythia 2.8b-deduped has different weights but is from the same model family as our target model, i.e., Pythia 410m. Consequently, for PeS2o, we use the OLMo-1b model to investigate the effect of using a different model architecture between the scoring and target model.
3. **Target model**: This is the model that we suspect has been trained using D'. We use the Pythia 410m model.

We continue pretraining the Pythia 410m model [3]. In addition to the watermarked text, we sample 5 billion tokens from the Common Pile dataset [13] for pretraining (See Appendix D for more details).

Baselines: We adopt five baselines as follows: **Maximum**: pick the paraphrase with the highest Min-K%++ score, **Random**: pick one of the paraphrases randomly. Additionally, **STAMP** and **LLM-DI** are described in Appendix F.

5.1 Evaluation of Watermarking

Table 2: p-values for different baselines compared against SPECTRA. Bold indicates a statistically significant result under a threshold of $p < 10^{-4}$. For true positives (TP) the p-value should be below this threshold and for false positives (FP), it should be above this threshold.

Method	PubMed			Wiki			HN			PeS2o		
	TP	FP	FP/TP	TP	FP	FP/TP	TP	FP	FP/TP	TP	FP	FP/TP
LLM-DI	0.06	**0.48**	7.67	0.02	**0.44**	22	0.49	**0.35**	0.71	0.02	**0.17**	8.50
STAMP	0.01	**0.48**	48	0.17	**0.03**	0.19	7E-4	**0.15**	214	0.15	**0.46**	3.07
Maximum	0.03	**1.00**	3.33	1.00	**1.00**	1	**3E-6**	**1.00**	3E5	0.95	**1.00**	1.05
Random	**1E-7**	**8E-4**	8E3	**5E-9**	2E-5	4E3	**4E-27**	**0.10**	3E25	1E-3	**0.11**	100
SPECTRA	**1E-17**	**0.02**	2E15	**4E-19**	**0.02**	5E16	**3E-60**	**0.59**	2E59	**2E-12**	**3E-3**	2E9

We measure statistical significance (p-values) for detecting the watermark in each dataset using the Pythia 410m model. Specifically, we compute true-positive p-values from the Pythia model trained on the watermarked datasets and false-positive p-values from the original Pythia 410m model that has not encountered the watermarked data during training.

We see from the results (Table 2) that SPECTRA is the *only one* that correctly detects membership for each dataset in the study under a threshold of $p < 10^{-4}$. Under a naïve approach of selecting paraphrases that maximize the Min-K%++ shift, the resulting pre-training shift is so large that subsequent training does not further amplify it, making pre- and post-training distributions indistinguishable and thus undetectable in practice. The random baseline also fails as Wiki results in false positives, while with PeS2o, it fails to detect a true positive. This indicates that the sampling strategy employed for paraphrases in SPECTRA is crucial to ensuring its performance. SPECTRA consistently has a high ratio of p-value between true positives and false positives ($> 10^9$). Notably, SPECTRA correctly detects membership for PeS2o, demonstrating that SPECTRA remains effective even as the scoring model and target model architecture differ. As suggested in Huh et al. [12], different LLMs follow similar training objectives, and as the amount of training data and tasks gets scaled up, the space of acceptable representations narrows dramatically, leading to similar learned representations. Additional results can be found in Appendix C, G, H.

In contrast, STAMP and LLM-DI fail to reliably detect true positives under our strict threshold ($p < 10^{-4}$). Although these methods achieve significance for certain datasets when adopting a more

permissive threshold $p < 0.05$, we argue that, due to the significant implications of falsely identifying datasets as training data in LLMs, a more stringent threshold is necessary and justified.

We computed the p-value of PeS2o on other LLMs as the target model that had a knowledge cut-off prior to the earliest released samples in PeS2o and thus could not have been trained on it (Table 3). All computed p-values are greater than the threshold of 10^{-4}. Thus, SPECTRA did not generate any false positives.

Table 3: p-value of PeS2o on models that have not seen it during pre-training. All models have $p > 10^{-4}$, indicating that membership could not be detected with statistical significance.

	Mistral-7b	GPT-Neo-2.7b	GPT-J-6b	Bloom-3b	Bloom-7b
p-value	0.08	0.04	0.003	0.33	0.02

5.2 Ablations on Scoring Model

We investigate the effect of using different scoring models for the ranking of rephrases. We aim to ascertain whether changing the scoring model would substantially affect the outcomes of SPECTRA. We utilize paraphrases derived from the PeS2o dataset. The OlMo-1b model was originally employed for scoring the PeS2o dataset in our experiments. We compare the rank ordering of paraphrases generated by OLMo-1b against Olmo-7b, Pythia 2.8b, Pythia 160m, and Pythia 6.9b. Each of these models, along with the original scoring model, is used to compute Min-K%++ scores for the paraphrases from the PeS2o dataset. The rankings generated by each model are then used to compute the Spearman [32] correlation scores between the original scoring model (OLMo-1b) and the additional scoring models.

Table 4: Rank-correlation coefficients between OLMo-1b and other scoring models.

	Olmo-7b	Pythia-2.8b-deduped	Pythia-160m-deduped	Pythia-6.9b
Spearman ρ	0.826	0.824	0.699	0.818

For three out of the four additional models, the Spearman's rank correlation coefficient (ρ) values exceed 0.8 (Table 4). This was true for models with more than 2.8b parameters, indicating that the correlation stabilizes for larger models. These findings suggest that SPECTRA is robust to changing the scoring model.

6 Conclusions

We presented SPECTRA, a watermarking approach that enables content creators to test if their data was used to train an LLM. Unlike previous approaches, SPECTRA does not need access to the decoding layer of a large LLM. SPECTRA also does not require access to a held-out dataset from the same domain that is typically necessary for MIA. We empirically show that SPECTRA can achieve a p-value gap of at least nine orders of magnitude between true positives and false positives, making this a reliable test of membership.

We highlight some important limitations of our study. We continue to pre-train the Pythia 410m model over a large number of tokens instead of training an LLM from scratch due to computational constraints. We assume access to the log probabilities output by the model, which can be challenging to achieve for proprietary models. Our approach will not help content creators who have already published their content, but can only help them going forward, as the watermarking must be done before publishing.

In terms of future directions, there is a need to develop watermarking techniques that can be used to verify membership not only by the content creator but also by interested third parties. Additional verification of our techniques at the scale of pre-training an LLM from scratch would inspire greater confidence that it can be employed in the event of any legal challenges.

Dislaimer

This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("JP Morgan") and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy, or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security,financial instrument, financial product, or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

[1] Mobina Alemi, Cristiana Gaiteiro, Carlos Alexandre Ribeiro, Luís Miguel Santos, João Rodrigues Gomes, Sandra Marisa Oliveira, Pierre-Olivier Couraud, Babette Weksler, Ignacio Romero, Maria João Saraiva, et al. Transthyretin participates in beta-amyloid transport from the brain to the liver-involvement of the low-density lipoprotein receptor-related protein 1? *Scientific reports*, 6(1):20164, 2016.

[2] Martin Bertran, Shuai Tang, Aaron Roth, Michael Kearns, Jamie H Morgenstern, and Steven Z Wu. Scalable membership inference attacks via quantile regression. *Advances in Neural Information Processing Systems*, 36:314–330, 2023.

[3] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[4] Wassim Bouaziz, Mathurin Videau, Nicolas Usunier, and El-Mahdi El-Mhamdi. Winter soldier: Backdooring language models at pre-training with indirect data poisoning. *arXiv preprint arXiv:2506.14913*, 2025.

[5] Blake Brittain. Authors sue Anthropic for copyright infringement over AI training. *Reuters*, August 2024. URL `https://www.reuters.com/technology/artificial-intelligence/authors-sue-anthropic-copyright-infringement-over-ai-training-2024-08-20/`. Technology/Artificial Intelligence section.

[6] 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)*, pages 2633–2650, 2021.

[7] Debeshee Das, Jie Zhang, and Florian Trantèr. Blind baselines beat membership inference attacks for foundation models. In *2025 IEEE Security and Privacy Workshops (SPW)*, pages 118–125. IEEE, 2025.

[8] 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? In *Conference on Language Modeling (COLM)*, 2024.

[9] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[10] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[11] Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the science of language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.841. URL https://aclanthology.org/2024.acl-long.841/.

[12] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Forty-first International Conference on Machine Learning*, 2024.

[13] Nikhil Kandpal, Brian Lester, Colin Raffel, Sebastian Majstorovic, Stella Biderman, Baber Abbasi, Luca Soldaini, Enrico Shippole, A Feder Cooper, Aviya Skowron, et al. The common pile v0. 1: An 8tb dataset of public domain and openly licensed text. *arXiv preprint arXiv:2506.05209*, 2025.

[14] 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. *arXiv preprint arXiv:2306.04634*, 2023.

[15] 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*, 36:27469–27500, 2023.

[16] Xianzhi Li, Ran Zmigrod, Zhiqiang Ma, Xiaomo Liu, and Xiaodan Zhu. Fine-tuning language models with differential privacy through adaptive noise allocation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8368–8375, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.491. URL https://aclanthology.org/2024.findings-emnlp.491/.

[17] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[18] Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Walsh, Yanai Elazar, Kyle Lo, et al. Paloma: A benchmark for evaluating language model fit. *Advances in Neural Information Processing Systems*, 37: 64338–64376, 2024.

[19] Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset?, 2024.

[20] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.719. URL https://aclanthology.org/2023.findings-acl.719/.

[21] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8332–8347, 2022.

[22] Yonatan Oren, Nicole Meister, Niladri S Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[23] Michael-Andrei Panaitescu-Liess, Zora Che, Bang An, Yuancheng Xu, Pankayaraj Pathmanathan, Souradip Chakraborty, Sicheng Zhu, Tom Goldstein, and Furong Huang. Can watermarking large language models prevent copyrighted text generation and hide training data? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25002–25009, 2025.

[24] Haritz Puerto, Martin Gubri, Sangdoo Yun, and Seong Joon Oh. Scaling up membership inference: When and how attacks succeed on large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4165–4182, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.234. URL https://aclanthology.org/2025.findings-naacl.234/.

[25] Saksham Rastogi, Pratyush Maini, and Danish Pruthi. Stamp your content: Proving dataset membership via watermarked rephrasings, 2025. URL https://arxiv.org/abs/2504.13416.

[26] Sruly Rosenblat, Tim O'Reilly, and Ilan Strauss. Beyond public access in llm pre-training data. *arXiv preprint arXiv:2505.00020*, 2025.

[27] Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. *Advances in Neural Information Processing Systems*, 37: 21079–21113, 2024.

[28] Tom Sander, Pierre Fernandez, Saeed Mahloujifar, Alain Oliviero Durmus, and Chuan Guo. Detecting benchmark contamination through watermarking. In *The 1st Workshop on GenAI Watermarking*, 2025.

[29] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models, 2023.

[30] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[31] Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

[32] Charles Spearman. The proof and measurement of association between two things. *International journal of epidemiology*, 39(5):1137–1150, 2010.

[33] Jonathan Stempel. NY Times sues OpenAI, Microsoft for infringing copyrighted work. *Reuters*, December 2023. URL https://www.reuters.com/legal/transactional/ny-times-sues-openai-microsoft-infringing-copyrighted-work-2023-12-27/. Legal/Transactional section.

[34] Johnny Wei, Ryan Wang, and Robin Jia. Proving membership in LLM pretraining data via data watermarks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13306–13320, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.788. URL https://aclanthology.org/2024.findings-acl.788/.

[35] John Wieting and Kevin Gimpel. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, 2018.

[36] John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-kirkpatrick. Paraphrastic representations at scale. In Wanxiang Che and Ekaterina Shutova, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 379–388, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-demos.38. URL https://aclanthology.org/2022.emnlp-demos.38/.

[37] Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. ReCaLL: Membership inference via relative conditional log-likelihoods. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8671–8689, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.493. URL https://aclanthology.org/2024.emnlp-main.493/.

[38] Jun Yan, Vansh Gupta, and Xiang Ren. Textual backdoor attacks with iterative trigger injection. *arXiv preprint arXiv:2205.12700*, 2022.

[39] 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)*, pages 268–282. IEEE, 2018.

[40] Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-k%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZGkfoufDaU.

[41] 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. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5263–5274, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.300. URL https://aclanthology.org/2024.emnlp-main.300/.

[42] Jing Zhao, Xiaoning Wang, Huachao Zhu, Suhua Wei, Hailing Zhang, Le Ma, and Pengcheng He. Integrative analysis of bulk rna-seq and single-cell rna-seq unveils novel prognostic biomarkers in multiple myeloma. *Biomolecules*, 12(12):1855, 2022.

A Datasets: Additional details

For each domain in the Pile that we use, we sample 500 documents for watermarking, and 500 are kept as held-out non-member data that are not used for training and are instead used as the validation set for LLM-DI. We pick the datasets from Duan et al. [8] that were deduplicated against the Pile training data with a 13-gram Bloom filter and a threshold of 80 % overlap. This means that the dataset was constructed from the held-out set of the Pile to avoid any document with up to a 13-gram overlap with any document included in the training set. Each training document was limited to between 100 and 200 words and at most 512 tokens. We control for the length as longer sequences make the document more detectable to MIA methods [24].

PeS2o: We use the PeS2o subset of Dolma available in the Paloma dataset [18]. To further deduplicate it, we compare the 'ID' field of this data against the entire Dolma dataset and remove all documents from Paloma whose 'ID' was found in Dolma. This left us with 432 documents, which we truncated as above to 512 tokens with between 100 and 200 words. We further split this into 216 documents used for watermarking and 216 documents used as held-out data.

The number of tokens in each dataset is shown in Table 5

Table 5: Number of tokens for each dataset computed using the Pythia tokenizer

Dataset	Number of tokens
Pubmed Central	154387
Wikipedia	141450
Hackernews	160637
PeS2o	56770

B Paraphraser LLM

We sample 10 unique paraphrases using the Llama-3.1-405b model using the prompt provided below. Each paraphrase is sampled at a different temperature value. We verify that each paraphrase is unique after lower-casing it. For cases where the generated paraphrase matches a previously generated paraphrase for the same document, we repeat the generation process until a unique paraphrase is obtained. Llama-3.1-405b was used through the Bedrock API.

SPECTRA Paraphrasing Prompt

Paraphrase the below paragraph of text enclosed in <text>tags, hereafter referred to as the original text. The paraphrased text should use different vocabulary, sentence structure, and style while preserving the meaning and tone of the original text. Do not remove any information from the original text while paraphrasing. Do not add any new information to the paraphrased text that is not present in the original text. Do not add any interpretive language to the paraphrased text that is not implied by the original text. Ensure that all technical details, findings, results, and other information such as tense, voice, and line breaks are preserved. Format your response as: PARAPHRASED PARAGRAPH: [your rephrased version] Based on the aforementioned directions, paraphrase the following text. <text >{text}</text >

C Validating Quality of Paraphrases

Table 6: P-SP scores on paraphrasing quality. P-SP scores measure how well the paraphrase preserves the semantic content of the original document.

	PubMed	**Wiki**	**HN**	**PeS2o**
P-SP	0.88	0.93	0.76	0.93

We validate the quality of paraphrasing by using the P-SP metric [36]. The P-SP metric is widely used to measure paraphrasing quality [25, 15]. For a human-generated paraphrase, the average P-SP is 0.78 [15]. Except for Hackernews, all watermarked datasets had a P-SP score above 0.88 (Table 6). The lower scores for Hackernews are explored next.

Human evaluation. We randomly select 54 watermarked documents from our datasets and distribute them among four evaluators. This distribution ensures that each evaluator reviews 27 documents, with each document being assessed by two different evaluators. The evaluation focuses on whether the paraphrasing preserves the (i) meaning, (ii) structure, and (iii) author tone of the original text. Evaluators evaluate on a Likert scale [17] of 1-5, with 5 being the best and 1 the worst.

While the mean scores for all three criteria exceeded 4, the scores for the structure preservation criterion are comparatively lower (Figure 3). Specifically, for conversational-style text such as on Hackernews, the paraphraser LLM occasionally fails to maintain the original structure (Figure 4).

Instructions for Human Evaluation

Please rate the paraphrased text based on the following three criteria.
1. Meaning Preservation, i.e., Rate higher if the meaning of the original text is better preserved. Please do not consider emotional tone in this rating.
2. Structural Preservation, i.e., Rate higher if the structure of the original text is better preserved.
3. Emotional Tone Preservation, i.e., Rate higher if the tone of the original text is better preserved.
Please score on a scale of 1-5, with 1 being the worst and 5 being the best.

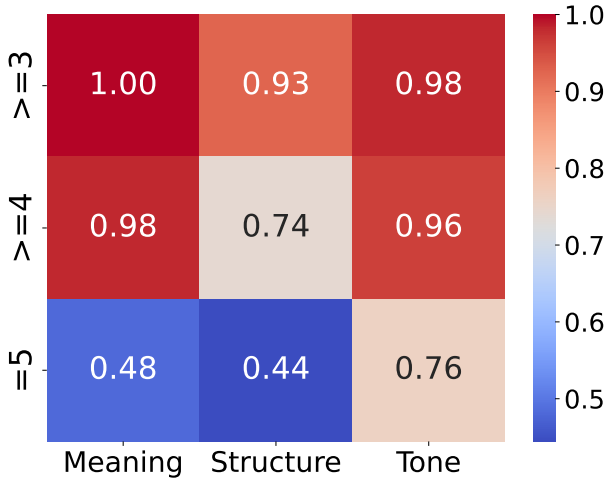

Figure 3: Heatmap showing fraction of evaluator scores for which the paraphrasings received a rating ≥ 3.

C.1 P-SP metric

The P-SP (PARAGRAM-SP) metric is used to quantify the semantic similarity between an original text and its paraphrase [36]. P-SP is computed as the cosine similarity between embeddings of the original and paraphrased texts, where each embedding is obtained by averaging the sentencepiece token embeddings of the respective text. In this metric, the only learned parameters are the sentencepiece embeddings themselves, which are trained on the ParaNMT paraphrase corpus [35].

D Continued Pretraining

In addition to the watermarked datasets, we sample documents from the Common Pile dataset for continued pre-training of Pythia 410m. Specifically, we select documents from the USPTO, USGPO, ArXiv, LibreText, and Doab domains of the Common Pile [13]. Each document is split into sequences of 512 tokens and used for training after randomizing the order. To avoid overlap with data seen during initial pre-training, we only sample documents published after December 2020, which is the cut-off date of the Pile dataset.

We use the AdamW optimizer with a learning rate of 10^{-4}, $(\beta_1, \beta_2) = (0.99, 0.999)$, cosine decay, and a batch size of 40. A warmup of 0.5% of training tokens was used with no weight decay. We train all our models on an L40S Tensor Core GPU. We utilize the Transformers library (version 4.43). We use a random seed of 1234 for all our experiments.

E Additional Details of MIA methods

DC-PDD: This method quantifies the divergence between the token log probabilities predicted by a target model and the empirical probabilities of those tokens in a reference corpus. Formally, we compute $P_M(x_t \mid x_{<t})$ where $P_M(x_t \mid x_{<t})$ is the model's predicted probability distribution for token x_t and $Q_{\text{ref}}(x_t)$ where $Q_{\text{ref}}(x_t)$ is the marginal token-frequency distribution estimated from the reference corpus, independent of context. $Q_{\text{ref}}(x_t)$ is pre-computed by adding the counts of each token in the vocabulary and computing its frequency against the total number of tokens in the corpus.

For each document $x = (x_1, \ldots, x_n)$, DC-PDD computes a calibrated score by measuring the divergence between the model's token-probability distribution and the reference token-frequency

Original and Paraphrased Text example for Hackernews dataset

Original

How to use Google Alerts to find out if your site gets hacked - greg http://www.blogstorm.co.uk/how-to-use-google-alerts-to-find-out-if-your-site-gets-hacked/

====== pierrefar It's a good backup solution as by the time Googlebot finds the cracked pages, it's a bit too late.

I'd set up a cron job to check the database once a day or 12 hours or so. That's more pre-emptive.

ZeroGravitas I think you're missing the point. These SEO hackers don't announce themselves by defacing your front page or deleting your database. They insert hidden links in your HTML to boost their PageRank and it's in their interests to go undetected for as long as possible.

pierrefar I'm an SEO and know full well what they're doing. My daily cron job suggestion is to detect their activities.

—— peregrine You would need a slightly larger set of words, but nonetheless an interesting solution.

streety Absolutely. Perhaps harvest words from blog spam.

Paraphrased

To detect potential hacking of your website, consider utilizing Google Alerts. However, by the time Googlebot discovers compromised pages, it may be too late. A more proactive approach would be to set up a scheduled task to check the database regularly, such as daily or every 12 hours. This method can help identify malicious activities before they cause significant damage. It's important to note that SEO hackers often insert hidden links in HTML to boost their PageRank, aiming to remain undetected for as long as possible. To combat this, a daily database check can be an effective solution. Additionally, using a larger set of words, possibly harvested from blog spam, can enhance the detection process.

Figure 4: Original and Paraphrased Texts from the Hackernews dataset: Exploring Methods to Detect Website Hacking Using Google Alerts. In this example, the paraphrased example does not follow the conversational format of the original text.

distribution. Specifically, it evaluates the following cross-entropy score:

$$-\frac{1}{|FOS(x)|}\sum_{t\,:\,\text{first occurrence}} P_M(x_t \mid x_{<t}) \cdot \log Q_{\text{ref}}(x_t)$$

where $FOS(x)$ denotes the set of tokens in x at their first occurrence. Only the first occurrence of each token in x is considered to reduce bias from repeated exposures.

For Min-k and Min-K%++ , we used K=20 %, which is the standard setting used in Duan et al. [8]. We used the Mimir library (https://github.com/iamgroot42/mimir) to compute all MIA scores.

F Watermarking baselines

1. **STAMP** [25]: Text is watermarked using the KGW scheme by rephrasing it with the Llama-70B model. This rephrasing is the public version. The same text is also watermarked using several other keys that are kept private. The perplexity is computed using the target model for all rephrases. The perplexity of the public watermarked text sequence is compared against the average perplexity of the private watermarked sequences over the dataset using a paired t-test. If the watermarked text is used for training the target model, then it is expected

to have lower perplexity than the average of the private rephrases. The implementation provided by the authors was used in our study.

In the evaluation of STAMP, the authors mainly considered Question Answering (QA) datasets; hence, in the prompt, the wording 'rephrase the question' is used. As we do not use QA datasets, we have modified the prompt to 'rephrase the text'.

2. **LLM-DI**: LLM-DI [19] utilizes outputs from multiple MIA methods as features to train a classifier for detecting training data. In this setting, the content creator retains an additional unreleased dataset. To ascertain whether a model has been trained on a released dataset, the creator must generate features from both the released and unreleased datasets and train a classifier. In our setup, we used the original documents (prior to watermarking) from each dataset as the released data and an equal number of held-out non-member documents from the same dataset as the unreleased data. The target model was trained using the original datasets without any watermarking. The implementation provided by the authors of LLM-DI was utilized in our study.

STAMP Paraphrasing Prompt

Rephrase the text given below. Ensure you keep all details present in the original, without omitting anything or adding any extra information not present in the original text.
Text: "'text"'
Your response should end with Rephrased Text: [rephrased text]

G Additional results

G.1 Watermarking Evaluation via Membership Inference Attacks

In addition to using p-values for watermark evaluation in the main paper, we report here the results of evaluating watermarking through a conventional Membership Inference Attack (MIA) framework. In this setting, for each dataset, we compare scores computed for documents that were used to train the model ("member" data) against scores for documents that were not included in training ("non-member" data). These scores are then used as inputs to a binary classifier that predicts whether a document is a member or a non-member. Member and non-member data are expected to have separable score distributions.

Specifically, for the Pythia 410 model trained with watermarked data, we calculate the area under the Receiver Operating Characteristic curve (ROC-AUC) and the true positive rate (TPR) at a fixed false positive rate (FPR) of 1%. High ROC-AUC values indicate strong separation between member and non-member data, while high TPR at low FPR measures the ability to detect true positives with minimal false positives.

As shown in Table 7, Min-K%++ achieves the highest ROC-AUC on three out of four datasets. However, as indicated in Table 8, even when ROC-AUC is high, reliably detecting members at a very low false positive rate remains challenging, highlighting a practical limitation of MIA.

	PubMed	**HN**	**Wiki**	**PeS2o**
Min-K	0.653	0.810	0.691	0.684
Loss	0.564	0.728	0.669	0.636
DC-PDD	0.663	**0.856**	0.709	0.710
Min-K%++	**0.743**	0.855	**0.761**	**0.718**

Table 7: ROC AUC of member data against non-member data computed using Pythia 410m model trained on watermarked data

G.2 Effect of Scoring Model Choice on Statistical Testing

In our experiments, we used the OLMo-1b model as the scoring model for watermarking the PeS2o dataset and for computing p-values via statistical testing on a target model. In Section 5.1, we

	PubMed	HN	Wiki	PeS2o
Min-K	4.4	16.4	3.4	2.3
Loss	0.8	3.8	1.6	1.9
DC-PDD	4.8	**27.2**	4.0	5.1
Min-K%++	**7.2**	24.0	**4.2**	**9.3**

Table 8: True Positive Rate (%) at False Positive Rate 1 % computed using Pythia 410m model trained on watermarked data

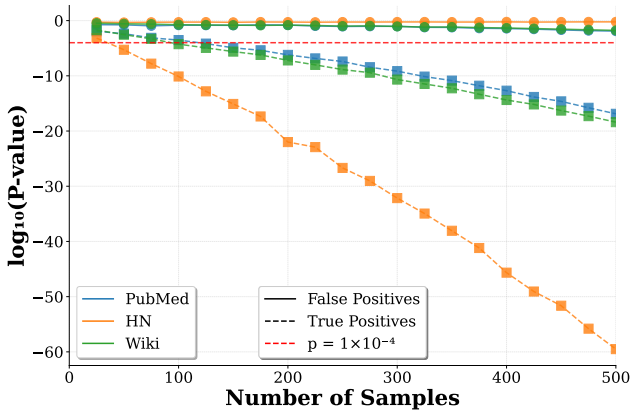

Figure 5: Trend of p-values with number of samples

investigated whether changing the scoring model during watermarking affected the ranking of paraphrases, finding that the order remained largely consistent across models. Here, we extend this analysis to examine the effect of varying the scoring model used during the statistical test itself. None of the scoring models employed for evaluation were exposed to the PeS2o dataset during their respective pre-training. As shown in Table 9, in all cases, PeS2o is detected with statistically significant p-values when it was included in the training data for the target model (i.e., Pythia 410m, representing true positives) Membership was not detected with statistical significance ($p < 10^{-4}$) when using the original Pythia 410m model (not trained on watermarked data) using any of the scoring models. This demonstrates the robustness of our approach to varying the scoring model in the statistical test.

Table 9: Effect of varying the scoring model for PeS2o

Scoring model	True positive	False positive
OLMo-7b	3E-5	0.56
Pythia-2.8b-deduped	2E-7	0.08
Pythia-6.9b-deduped	1E-5	0.59
OLMo-1b	2E-12	3E-3

H Additional Ablation studies

H.1 Number of Samples

The p-value for any of the datasets we test goes below the threshold after 100-150 samples, suggesting that this is the minimum number of samples needed. The p-value for false positives is always above the threshold (Figure 5).

H.2 Effect of training tokens on p-value

For all four datasets, the p-value remains well below the significance threshold $p < 10^{-4}$ throughout the training run (Figure 6). This indicates that our t-test would be effective at any point during

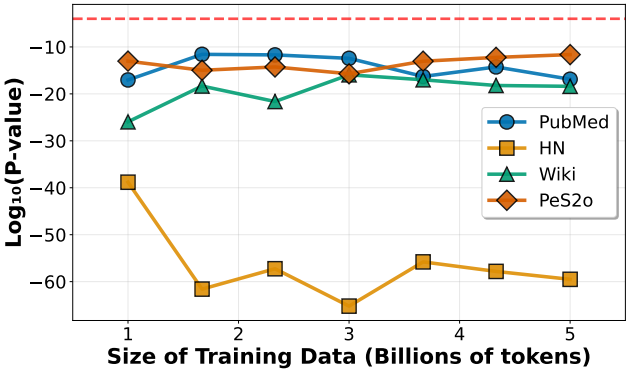

Figure 6: Trend of p-values with the number of pre-training tokens. The dashed red line indicates the p-value threshold of 10^{-4}

training. After 3.5 billion training tokens, the p-values show little further change, suggesting that the results remain stable even as training progresses beyond this point.

H.3 Evaluating Alternative MIA Scores in SPECTRA

We replace the Min-K%++ score in SPECTRA with other MIA scores and report the resulting p-value performance in Table 10. Most alternative methods either tend to produce false positives (such as min-k and loss) or fail to reliably detect true positives (such as DC-PDD).

Table 10: p-values for different scoring methods compared against SPECTRA. $p < 10^{-4}$ for membership to be detected with statistical significance. For all datasets, methods other than Min-K%++ are unable to either detect or reject membership, showing a consistent bias for one or the other depending on the method. TP refers to true positives, and FP refers to false positives. The last row corresponds to SPECTRA

Method	**PubMed**		**Wiki**		**HN**		**PeS2o**	
	TP	FP	TP	FP	TP	FP	TP	FP
Loss	1E-73	4E-13	4E-34	6.2E-20	6E-178	1E-53	7E-23	7E-11
DC-PDD	1.0	1.0	1.0	0.98	1.0	0.49	1.0	1.0
min-k	2E-74	2.8E-10	1.6E-34	1E-17	1E-169	3.3E-12	2E-21	4E-7
Min-K%++	1E-17	0.02	4E-19	0.02	3E-60	0.59	2E-12	3E-3

I Extended Related Work

Membership Inference Attack Membership inference attacks (MIA) were first systematically studied for simpler neural and convolutional networks by Shokri et al. [30]. Their approach involves training multiple shadow models, each on different non-overlapping subsets of data drawn from the same distribution as the target model's training set. The shadow models generate features used to train an attack model that predicts membership status. However, this shadow model approach is computationally infeasible for large language models (LLMs) with billions of parameters.

Subsequent work has focused on more scalable membership inference methods. For example, Mattern et al. [20] proposed the Neighborhood Attack, which perturbs a document by masking parts of its text and replacing the masked tokens with predictions from a BERT-like model ("neighbor"). The MIA score is defined as the difference between the loss on the original document and the average loss across its perturbed neighbors; training documents typically yield lower values than non-members.

The MIA techniques described thus far do not require a reference model. In contrast, reference-based attacks such as LiRA (Likelihood Ratio Attack) [21] do require a reference model trained on data from a similar, but largely non-overlapping, distribution. LiRA measures membership by comparing

the loss of the target model to that of the reference model on the same document, thereby calibrating for the intrinsic difficulty of the document. As observed by [8], obtaining a suitable reference model for LiRA can be challenging in practice.

Other recent works have shown that knowledge of non-member data can be leveraged to calibrate detection. [2] use quantile regression on the likelihoods of non-member data (computed with the target model) to estimate a threshold separating members from non-members. Unlike earlier work, which often focuses on detecting pre-training data (where the model sees each example only once), Bertran et al. target fine-tuning data, which may be seen multiple times during training.

More recently, the ReCALL method prepends a prefix of non-member data to member documents and computes the ratio of the document likelihood with and without the prefix. They demonstrate that this likelihood ratio changes more for member documents than for non-members [37].

While the term “attack" is often used in an adversarial context, denoting unauthorized detection of training data, in our work, we repurpose MIA techniques as tools for data transparency—enabling enforcement of data use policies and verification of model training practices.

Training Data Privacy A related line of work, known as training data extraction, seeks to prove membership by generating verbatim training examples from a trained model [6]. However, successful extraction typically requires knowledge of effective prefixes or prompts and a reliable method for verifying whether the generated output indeed originated from the training set. Moreover, not all data used to train a model is equally susceptible to extraction, as memorization varies widely across training examples.

Differential privacy is a technique designed to limit the information leakage of individual training data points. This is usually achieved by clipping gradients on a per-example basis, thus bounding the contribution of any single example to the model parameters. However, differential privacy methods often result in reduced downstream performance and are rarely adopted in large language model (LLM) training, since per-example gradient computation significantly increases training costs [16].

Beyond membership inference attacks, alternative signals have been proposed to detect whether a particular dataset was used in model training. For example, Oren et al. [22] investigate whether a model assigns higher likelihood to the canonical ordering of documents in a dataset compared to other permutations. This may indicate exposure to the dataset during training. However, this method relies on the assumption that data are presented in canonical order during training, which may not hold in practice due to common data shuffling procedures.

Watermarking LLM output While our work focuses on watermarking training data, a large body of literature addresses watermarking the output of LLMs to detect whether an LLM generated a given text. A popular statistical watermarking technique partitions the model’s vocabulary into a “red list” and a “green list,” and increases the logits of the green list tokens by a constant δ during generation. The presence of a watermark is then detected by measuring the frequency of green list tokens in the output and applying a statistical test, such as a z-test [14].

Sander et al. [27] demonstrated that watermarked text is “radioactive,” meaning it leaves a detectable trace if the watermarked text is later used to train another model. Sander et al. [28] extended this technique to detect benchmark contamination in LLMs, but found that successful detection required the contaminated data to be repeated many times in the pretraining corpus (at least 16 times). The STAMP paper [25] reported that the statistical test from Sander et al. [28] was unable to detect training data when it appeared fewer times in the corpus. Additionally, Panaitescu-Liess et al. [23] found that watermarking text can diminish the effectiveness of membership inference attack (MIA) techniques.

J Watermarking Examples

We provide an example from each of our datasets of the watermarked and original document for SPECTRA in Figure 2-5.

Original and Paraphrased Text example for Wikipedia dataset

Original

Charles Theodore ‘Theo’ Harding (26 May 1860 – 13 July 1919) was an English-born international rugby union player who played club rugby for Newport and international rugby for Wales. Harding was an all-round sportsman and also captained Newport Hockey Club in their very first season.

Rugby career

Harding was one of the first Newport players and was given the captaincy of the club in the 1887/88 and the 1888/89 season. During the 1888 Harding twice faced the first overseas touring team the New Zealand Māoris. The first occasion was also Harding’s first cap for Wales, when under the captaincy of Frank Hill, the Welsh team beat the tourists five points to nil. Four days later, on 26 December, Harding led his Newport team against the Māori’s, but without star player and Welsh legend Arthur Gould, Newport’s supporters were not optimistic of success. They were proven right when the Māoris won three tries to nil.

In 1889, Harding was selected to represent Wales twice as part of the Home Nations Championship. Wales lost both games of the tournament against Scotland and Ireland, and Harding was not chosen to represent his country again.

International matches played

Wales (rugby union)

1889

Paraphrased

Charles Theodore "Theo" Harding (May 26, 1860 - July 13, 1919) was a renowned English-born rugby union player who showcased his skills at both club and international levels, representing Newport and Wales, respectively. Harding’s versatility as an athlete extended beyond rugby, as he also led Newport Hockey Club to success in their inaugural season.

Rugby Career

As one of Newport’s pioneering players, Harding assumed the role of captain for two consecutive seasons (1887/88 and 1888/89). During this period, he encountered the first-ever overseas touring team, the New Zealand Māoris, on two separate occasions. The initial encounter marked Harding’s debut appearance for Wales, where, under the leadership of Frank Hill, the team emerged victorious with a score of five points to nil. Just four days later, on December 26, Harding led his Newport team against the Māoris, but the absence of the legendary Arthur Gould dampened the spirits of Newport’s supporters, who correctly predicted a defeat, as the Māoris secured three tries to nil.

In 1889, Harding was selected to represent Wales in two Home Nations Championship matches. Unfortunately, Wales suffered losses against both Scotland and Ireland, and Harding did not receive further international call-ups.

International Matches Played

Wales (Rugby Union)

1889

Figure 7: Comparison of Original and Paraphrased Texts from the Wikipedia dataset: An Example Featuring Charles Theodore ‘Theo’ Harding’s Rugby Career.

Original and Paraphrased Text example for PubMed Central dataset

Original

Alzheimer's disease (AD), described for the first time by Alois Alzheimer in 1906, is characterized by progressive loss of cognitive functions ultimately leading to death[@b1]. Pathologically, the disease is characterized by the presence of extraneuronal amyloid plaques consisting of aggregates of amyloid-beta (Aβ) peptide, and neurofibrillary tangles (NFTs) which are intracellular aggregates of abnormally hyperphosphorylated tau protein[@b2]. Aβ peptide is generated upon sequential cleavage of the amyloid precursor protein (APP), by beta- and gamma-secretases, and it is believed that an imbalance between Aβ production and clearance results in its accumulation in the brain.

Clearance of Aβ from the brain occurs via active transport at the blood-brain-barrier (BBB) and blood cerebrospinal fluid (CSF) barrier (BCSFB), in addition to the peptidolytic removal of the peptide by several enzymes. The receptors for Aβ at the BBB bind Aβ directly, or bind to one of its carrier proteins, and transport it across the endothelial cell. The low-density lipoprotein receptor-related protein 1 (LRP1) and the receptor for advanced glycation end products (RAGE) are involved in receptor-mediated flux of Aβ across the BBB[@b3]. Both LRP1 and RAGE are multi-ligand cell surface receptors that, in addition to Aβ, mediate the clearance of a large number of proteins.

Paraphrased

Alzheimer's disease, first identified by Alois Alzheimer in 1906, is marked by a gradual decline in cognitive abilities, ultimately resulting in mortality[@b1]. From a pathological perspective, the disease is distinguished by the presence of extraneuronal amyloid plaques comprising aggregated amyloid-beta peptides, as well as neurofibrillary tangles, which are intracellular accumulations of aberrantly hyperphosphorylated tau protein[@b2]. The amyloid-beta peptide is generated through the sequential cleavage of the amyloid precursor protein by beta- and gamma-secretases, and it is hypothesized that an imbalance between its production and clearance leads to its accumulation in the brain.

The removal of amyloid-beta from the brain occurs through active transport at the blood-brain barrier and blood-cerebrospinal fluid barrier, in addition to enzymatic degradation by various peptidases. At the blood-brain barrier, receptors for amyloid-beta bind the peptide directly or interact with one of its carrier proteins, facilitating its transport across the endothelial cell. The low-density lipoprotein receptor-related protein 1 and the receptor for advanced glycation end products play a role in receptor-mediated amyloid-beta flux across the blood-brain barrier[@b3]. Both of these receptors are multifunctional cell surface receptors that, in addition to amyloid-beta, mediate the clearance of numerous other proteins.

Figure 8: Comparative Analysis of Original and Paraphrased Texts from the PubMed Central dataset [1].

Original and Paraphrased Text example for PeS2o dataset

Original

Molecular heterogeneity has great significance in the disease biology of multiple myeloma (MM). Thus, the analysis combined single-cell RNA-seq (scRNA-seq) and bulk RNA-seq data were performed to investigate the clonal evolution characteristics and to find novel prognostic targets in MM. The scRNA-seq data were analyzed by the Seurat pipeline and Monocle 2 to identify MM cell branches with different differentiation states. Marker genes in each branch were uploaded to the STRING database to construct the Protein-Protein Interaction (PPI) network, followed by the detection of hub genes by Cytoscape software. Using bulk RNA-seq data, Kaplan-Meier (K-M) survival analysis was then carried out to determine prognostic biomarkers in MM. A total of 342 marker genes in two branches with different differentiation states were identified, and the top 20 marker genes with the highest scores in the network calculated by the MCC algorithm were selected as hub genes in MM. Furthermore, K-M survival analysis revealed that higher NDUFB8, COX6C, NDUFA6, USMG5, and COX5B expression correlated closely with a worse prognosis in MM patients.

Paraphrased

A comprehensive analysis integrating bulk RNA sequencing and single-cell RNA sequencing has shed light on novel prognostic biomarkers in multiple myeloma, a disease characterized by significant molecular heterogeneity. To investigate clonal evolution and identify potential targets, researchers combined single-cell RNA sequencing data, analyzed using the Seurat pipeline and Monocle 2, with bulk RNA sequencing data. This approach enabled the identification of distinct branches of multiple myeloma cells with varying differentiation states. The marker genes associated with each branch were used to construct a Protein-Protein Interaction network via the STRING database, and hub genes were detected using Cytoscape software. Subsequent Kaplan-Meier survival analysis, utilizing bulk RNA sequencing data, revealed prognostic biomarkers for multiple myeloma. A total of 342 marker genes were identified across two branches with differing differentiation states, and the top 20 genes with the highest scores, as calculated by the MCC algorithm, were selected as hub genes. Furthermore, survival analysis demonstrated that elevated expression of NDUFB8, COX6C, NDUFA6, USMG5, and COX5B correlated strongly with a poorer prognosis in multiple myeloma patients.

Figure 9: Original and Paraphrased Texts from the PeS2o dataset [42]