Detecting Non-Membership in LLM Training Data via Rank Correlations

Pranav Shetty, Mirazul Haque, Zhiqiang Ma, Xiaomo Liu
JPMorgan AI Research
first.last@jpmchase.com

Abstract

As large language models (LLMs) are trained on increasingly vast and opaque text corpora, determining which data contributed to training has become essential for copyright enforcement, compliance auditing, and user trust. While prior work focuses on detecting whether a dataset *was* used in training (membership inference), the complementary problem—verifying that a dataset was *not* used- has received little attention. We address this gap by introducing PRISM, a test that detects dataset-level *non-membership* using only grey-box access to model logits. Our key insight is that two models that have not seen a dataset exhibit higher rank correlation in their normalized token log probabilities than when one model has been trained on that data. Using this observation, we construct a correlation-based test that detects non-membership. Empirically, PRISM reliably rules out membership in training data across all datasets tested while avoiding false positives, thus offering a framework for verifying that specific datasets were excluded from LLM training.

1 Introduction

Large Language Models (LLMs) owe their broad general-purpose capabilities to training on massive, diverse corpora of high-quality text (Wettig et al., 2024). Much of this material is scraped from the open internet, raising complex legal and compliance issues for every stakeholder in the ecosystem—from content creators and publishers to model developers and end-users. Many publishers have filed legal challenges that their content has been used without authorization and compensation to train LLMs and face the burden of proving that their data has been used for training (Brittain, 2024; Stempel, 2023). On the flip side, LLM trainers face the opposite burden of proving that they have *not* used a given dataset for training. Similarly, enterprises onboarding LLMs for internal use must perform compliance audits of these models to understand the legal risk in using these models in case the models have been trained using proprietary datasets. Similar verification is essential for enterprises that adopt third-party LLMs, which must audit models for potential exposure to proprietary or restricted data to assess legal and compliance risk. Non-membership detection is equally critical when users or organizations wish to ensure that sensitive or legally protected datasets—shared under agreements restricting their use for training—have genuinely been excluded. As modern LLMs are increasingly trained on corpora approaching "the sum of all human text," the ability to reliably establish what a model has not seen becomes as important as identifying what it has.

Membership inference attacks (MIA) have been studied in the literature to assess whether a certain dataset has been used to train a model (i.e., is a member of the training data) (Shi et al., 2023; Zhang et al., 2024b). These MIA scores typically rely on analyzing the token probabilities and comparing the token probabilities of member data (i.e., used for training) against non-member data (not used during training) from the same distribution. Member data typically has a higher probability than similar data not seen during training. However, these methods were shown to overfit to distributional differences between member and non-member data and performed no better than random once these differences were corrected (Duan et al., 2024). To address this issue, LLM Dataset Inference (DI) was proposed (Maini et al., 2024). LLM-DI does not perform inference on each document in a dataset but instead performs inference on the entire dataset. As this reduces variance due to outliers, we adopt this setting in our work.

However, LLM-DI and previously proposed MIA techniques have two major limitations. Firstly, they require access to known non-member data from an identical distribution to detect member-

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5414–5429

ship. In practice, obtaining such a distributionally matched reference dataset is rarely feasible, especially for proprietary or domain-specific corpora. Second, existing methods are inherently one-sided: they are designed only to provide evidence for membership in the training data. To date, no method to the best of our knowledge explicitly addresses the complementary problem of non-membership detection, i.e., ruling out that a dataset was used during training. Methods like LLM-DI perform a one-tailed hypothesis test and can either detect that a dataset was used for training or fail to detect this. Mere failure to detect membership does not automatically imply evidence of non-membership. This asymmetry leaves open a crucial gap: how to confidently demonstrate that a model has not been trained on a given dataset.

To address these limitations, we propose Normalized token log **P**robability **R**ank correlation **I**nference using **S**pearman for non-**M**embership detection (PRISM). We use the Min-K%++ score as our signal for detecting membership. Min-K%++ computes the average of the lowest K token log probabilities in any document, normalized using the mean and variance over the model's vocabulary. We begin with a reference model known not to have been trained on the suspect dataset and a target model whose training history we wish to test. Our central insight is that the Spearman rank correlation between the Min-K%++ scores produced by these two models serves as a reliable indicator of dataset membership. If the target model has not been trained on the dataset, both models—being sufficiently capable language models—will rank documents according to their underlying linguistic difficulty, yielding a high correlation. Conversely, if the target model has been trained on the dataset, memorization effects distort this ranking, leading to a lower correlation.

Our core contributions are as follows:

1. We propose the task of dataset-level non-member detection and demonstrate that PRISM reliably detects non-membership across all datasets and models using only the logits of the model.

2. PRISM does not require access to any additional reference datasets that were known to be held out from training a model, which is required by existing methods and is challenging to obtain in practice (Zhao et al., 2025).

3. PRISM can be performed with as few as 100 documents, making it practical to use.

2 Related Work

Membership Inference Attack. Early research on membership inference attack (MIA) focused on simpler neural architectures and convolutional neural networks, and involved training multiple shadow models on various dataset partitions to extract features indicative of membership status (Shokri et al., 2017). However, this shadow model approach does not scale to LLMs due to their large parameter count and computational demands. To address this limitation, more recent work has shifted towards using the log probability scores produced by LLMs to distinguish between member and non-member data points (Zhang et al., 2024b; Shi et al., 2023).

To facilitate the evaluation of these log-probability-based MIA techniques, researchers introduced benchmark datasets such as WikiMIA (Shi et al., 2023) and PatentMIA (Zhang et al., 2024b). These datasets were curated by selecting documents published before the LLM's training cutoff as members and those published after as non-members. However, subsequent analysis revealed that the effectiveness of these attacks was largely attributable to temporal artifacts inherent in the dataset construction process. Temporal shifts lead to shifts in the specific language used in the member and non-member datasets, which is what these methods detected. When member and non-member data were drawn from the same distribution, the performance of all evaluated MIA methods dropped to no better than random, indicating that prior methods did not infer membership but rather they exploited these temporal signals (Duan et al., 2024; Maini et al., 2024; Das et al., 2025).

Dataset Inference. Unlike traditional membership inference attacks, which aim to determine the membership status of individual data points, Dataset Inference (DI) shifts the focus to assessing the likelihood that an entire dataset was included in the training process (Maini et al., 2024). Rather than producing granular, document-level predictions, DI seeks to estimate a confidence score for the entire dataset. This approach is motivated by the observation that LLM training pipelines often ingest data in bulk from specific sources, resulting in groups of related documents being consistently present in the training set. By aggregating signals across multiple

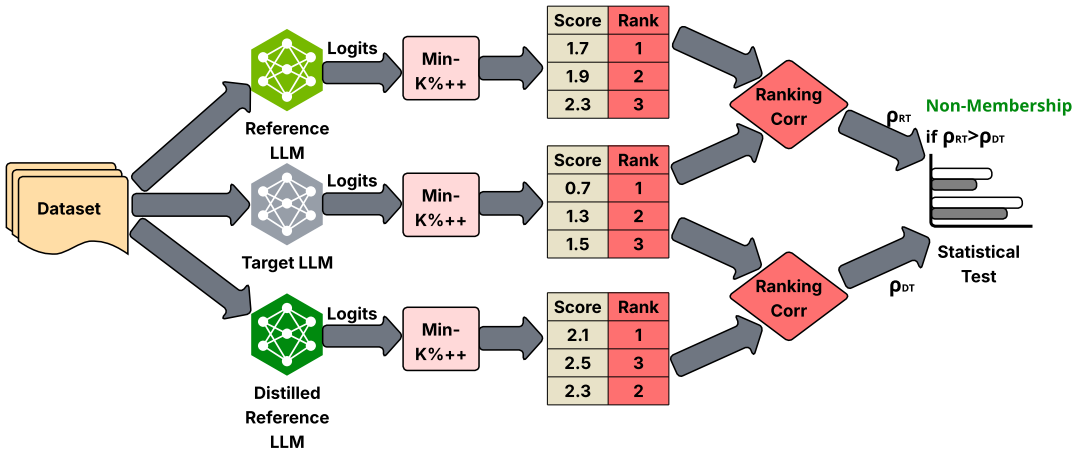


Figure 1: Overview of PRISM. Logits of Reference LLM, Target LLM, and Distilled Reference LLMs are used to calculate Min-K%++ scores, and the corresponding ranking of the scores. Final statistical testing for non-membership detection is dependent on the rank correlation between the reference and target model, and the rank correlation between the distilled reference and target model.

documents, DI not only amplifies the overall detection capability but also mitigates the impact of noisy or anomalous samples (Shetty et al., 2025).

3 Preliminaries

3.1 Problem setup

We consider the setting in which a verifier wishes to detect with high confidence whether a given suspect dataset $\mathcal{D}$ was *not* used in training a target LLM (non-membership detection), denoted by M_T. $\mathcal{D}$ is a collection of documents $\{x^{(i)}\}_{i=1}^{n}$, where each $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_{k_i}^{(i)})$ is a sequence of tokens drawn from a vocabulary $\mathcal{V}$. We assume *grey-box access* to M_T, meaning that for each token x_t in a sequence x, we can query the model to obtain its log probabilities over the known vocabulary $\mathcal{V}$. Formally, for a prefix $x_{<t}$, we may compute

$$\ell_{M_T}(x_t \mid x_{<t}) = \log P(x_t \mid x_{<t}; M_T),$$

where $P(\cdot \mid x_{<t}; M_T)$ denotes M_T's predicted distribution over $\mathcal{V}$ at step t. Crucially, we do not assume access to model parameters or gradients. This level of access is consistent with what is available for some commercial LLM APIs. Using model logits is more informative than a black-box approach but less so than using deeper layers of the model (Chen et al., 2025a). Our choice of a grey-box approach is motivated by practical applicability rather than downstream performance, as access to target model weights is often not feasible.

Reference LLM. We assume access to a reference LLM (M_R), which is open-source and known not to have been trained on $\mathcal{D}$. Such a model can often be identified by leveraging release dates: if the suspect dataset's public availability date is known, a model whose training data cutoff predates that release can be used as M_R. In practice, for many types of data, such as scientific literature, news articles, and online forums, establishing such a timestamp is feasible.

3.2 Min-K%++

Many scores have been proposed for membership detection, such as LOSS (Yeom et al., 2018), Min-k (Shi et al., 2023), and zlib (Carlini et al., 2021) (Appendix E). Min-K%++ (Zhang et al., 2025) focuses specifically on the $K\%$ of tokens that have the lowest normalized likelihood under a model. Min-K%++ incorporates normalization relative to the mean and variance of token log probabilities.

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

$$z(x_t; M) = \frac{\log P(x_t \mid x_{<t}; M) - \mu_{x_{<t}}}{\sigma_{x_{<t}}},$$

where

$$\mu_{x_{<t}} = \mathbb{E}_{z \sim P(\cdot \mid x_{<t}; M)}[\log P(z \mid x_{<t}; M)],$$
$$\sigma_{x_{<t}} = \sqrt{\mathrm{Var}_{z \sim P(\cdot \mid x_{<t}; M)}[\log P(z \mid x_{<t}; M)]}.$$

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. Var denotes the variance.

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 (Zhang et al., 2025) has shown that the Min-K%++ score theoretically corresponds to measuring the negative trace of the Hessian matrix of the log-likelihood $log\ P(x_t \mid x_{<t}; M)$. 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.

3.3 Empirical motivation for PRISM

To ground our approach, we begin with an empirical observation using the Pythia-410m model family. We compute the Min-K%++ scores of documents under both the original Pythia-410m model and a version that has undergone continued pretraining on additional data, referred to as Pythia-410m-CPT. Specifically, Pythia-410m-CPT was trained on several datasets drawn from the Pile validation split (listed in Table 1)—datasets that were not part of the original Pythia-410m training—and exposed to an additional six billion tokens of text.

We then measure the Spearman rank correlation of Min-K%++ scores between each model (Pythia-410m or Pythia-410m-CPT) and a set of independent reference models from the same Pythia family that have not seen these datasets. Across all datasets and reference models, we observe a consistent pattern: the rank correlation between the untrained Pythia-410m and each reference model is higher than that between the trained Pythia-410m-CPT and the same reference model. Among several membership-inference metrics we tested, Min-K%++ produced the most consistent and separable differences (see Appendix F).

This pattern suggests an intuitive mechanism. Two transformer models trained on large, overlapping web corpora but not exposed to the target dataset will tend to share similar priors about linguistic difficulty and token surprisal, leading them to rank documents in roughly the same order. When one of these models is later trained on that dataset, memorization effects perturb these priors and alter the ranking, thereby lowering the rank correlation.

This observation motivates our hypothesis: The relative rank correlation between a target model and a reference model can serve as a robust signal of dataset membership. In particular, a higher correlation with a reference model that has not seen the data—compared to one that has—provides strong evidence of non-membership. The next section formalizes this idea into a statistical test capable of detecting non-membership.

Table 1: Rank correlation coefficients over different datasets between Pythia reference models of different sizes computed against Pythia-410m and Pythia-410m-CPT. Each of the datasets was held out from training the vanilla model as well as each reference model in the columns, but was used during continued pretraining of Pythia-410m-CPT with 6 billion additional tokens.

Dataset	Pythia-1b		Pythia-2.8b		Pythia-6.9b	
	410m	410m-CPT	410m	410m-CPT	410m	410m-CPT
ArXiv	0.82	0.45	0.75	0.42	0.72	0.37
HN	0.71	0.40	0.74	0.40	0.68	0.41
CC	0.80	0.50	0.77	0.46	0.70	0.42
Wikipedia	0.84	0.25	0.80	0.25	0.77	0.21
PubMed	0.84	0.33	0.79	0.29	0.77	0.30

4 Methods

4.1 Comparing Rank Correlations

PRISM operates by comparing how the suspect dataset is ranked, in terms of difficulty, across three models: (1) the target model (M_T), (2) the reference model (M_R), and (3) a derived model called the distilled reference (M_D). Figure 1 illustrates this setup.

The distilled reference is created by fine-tuning the reference model on the suspect dataset while performing knowledge distillation from the target model. This dual training process uses the target model as a teacher—aligning the logits of the distilled reference with those of the target—while simultaneously training it on the suspect data. As a result, M_D approximates how the target model would behave if it had been trained on the dataset,

effectively serving as a proxy for "the target model exposed to $\mathcal{D}$."

PRISM computes Min-K%++ scores for all documents in the dataset using each of the three models and evaluates their Spearman rank correlations. If the target model has not been trained on the suspect data, its correlation with the reference model will exceed its correlation with the distilled reference, since only the latter has been exposed to $\mathcal{D}$. Conversely, if the target model has been trained on $\mathcal{D}$, it will align more closely with the distilled reference, resulting in higher correlation with it due to knowledge distillation. A crucial practical consideration is balancing the fine-tuning and distillation when training the distilled reference. Excessive distillation can make the model too similar to the target, obscuring the difference in rank correlations.

We use rank correlations here as they are robust to changes in model scale between the target model and scoring model. They are also robust to outliers (Schober et al., 2018; Tabatabai et al., 2021).

4.2 Training M_D

We obtain M_D by simultaneously fine-tuning on $\mathcal{D}$ and performing knowledge distillation using the logits of M_T to minimize the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) between the probability distributions of the target and reference models (Figure 2).

The loss used to obtain M_D from M_R is a linear combination of a supervised cross-entropy loss on the true tokens and a distillation term encouraging M_D to match the logits of M_T. Formally, during training, we minimize the following loss over $\mathcal{D}$.

$$\mathcal{L}(\theta) = (1-\lambda)\,\mathbb{E}_{x\sim\mathcal{D}} \sum_{t=1}^{|x|} \underbrace{\mathrm{CE}(P_{M_\mathrm{D}},\, \delta_{x_t})}_{\text{cross-entropy on ground truth}} + \lambda\tau^2\,\mathbb{E}_{x\sim\mathcal{D}} \sum_{t=1}^{|x|} \underbrace{\mathrm{KL}(P_{M_T} \parallel P_{M_\mathrm{D}})}_{\text{knowledge distillation}},$$

where, δ_{x_t} is the one-hot distribution on the observed token x_t, CE denotes the cross-entropy, and $\lambda \in [0,1]$ balances the two terms. θ are the parameters of M_D. Here P_M is used as short hand for $P_M(\cdot \mid x_{<t})$ for each model. Each probability distribution used in the KL term is obtained from the logits $\mathbf{z} \in \mathbb{R}^{|\mathcal{V}|}$ with $P_M = \mathrm{softmax}(\frac{\mathbf{z}}{\tau})$ where τ is the softmax temperature. The factor of τ^2 is

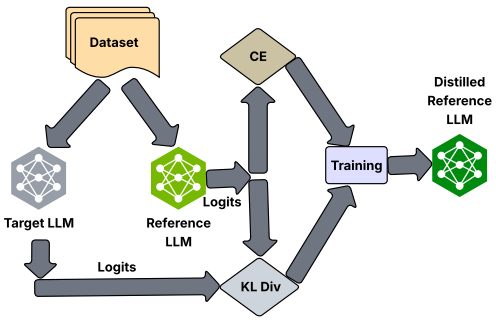


Figure 2: Overview of distilling the reference model. The distillation is dependent on two components: cross entropy of the reference model and KL divergence between the output probability of the target and reference model.

standard in knowledge distillation (Hinton et al., 2015) and is used to ensure the gradients from the cross-entropy and KL-divergence term are scaled similarly. We use $\lambda = 0.7$ and $\tau = 2$, which were obtained after a hyperparameter study (Appendix G). See Appendix C for more training details.

Knowledge distillation in this case only requires access to the logits of the target model. In general, the vocabularies of M_T and M_D may be different. When the vocabularies of the two models are different, it is necessary to align tokens when computing the KL divergence. This is a well-studied problem in the literature (Chen et al., 2025b; Boizard et al., 2024). We do not perform an explicit alignment, as the target and reference models in our study either share the same tokenizer or a similar one.

4.3 Hypothesis Testing via Rank-Correlation Comparisons

Define the Spearman rank correlation between the Min-K%++ scores ouput by models M_A and M_B over $\mathcal{D}$ as

$$\rho(M_A, M_B; \mathcal{D}) \equiv \mathrm{corr}\big(\mathrm{rank}(\{f_{\text{Min-K\%++}}(x^{(i)}; M_A)\}_{i=1}^n), \mathrm{rank}(\{f_{\text{Min-K\%++}}(x^{(i)}; M_B)\}_{i=1}^n)\big)$$

where $\mathrm{rank}(.)$ maps scores to ranks within $\mathcal{D}$ and $\mathrm{corr}(.)$ is the Pearson correlation coefficient operator. We use the short-hand notation ρ_{AB} for this.

Test statistic. Our method compares the target's rank agreement with the reference versus with the distilled reference. Let

$$\widehat{\Delta} \equiv \rho(M_R, M_T; \mathcal{D}) - \rho(M_D, M_T; \mathcal{D})\,.$$

$$\widehat{\Delta} \equiv \rho_{RT} - \rho_{FT}$$

Bootstrap inference. As we are comparing two dependent correlations, we use a bootstrap percentile to compute a p-value (Wilcox, 2016; Dhingra et al., 2019; Wilcox, 2017). We construct a test on $\widehat{\Delta}$ using nonparametric bootstrap over sequences:

1. For $b = 1, \ldots, B$ (with B=10,000), draw a bootstrap resample $\mathcal{D}^{(b)}$ of size n by sampling sequences from $\mathcal{D}$ with replacement.
2. Compute $\rho_{DT}^{(b)} = \rho(M_D, M_T; \mathcal{D}^{(b)})$ and $\rho_{RT}^{(b)} = \rho(M_R, M_T; \mathcal{D}^{(b)})$, and set $\Delta^{(b)} = \rho_{RT}^{(b)} - \rho_{DT}^{(b)}$.

$\{\Delta^{(b)}\}_{b=1}^{B}$ is known to approximate the sampling distribution of $\widehat{\Delta}$ (Efron, 1992).

Test for evidence of *non*-training on $\mathcal{D}$ (non-membership). The one-sided hypothesis test on the statistic $\widehat{\Delta}$ is as below where Δ is the true value estimated by $\widehat{\Delta}$:

$$H_0^{\text{non}} : \Delta \leq 0 \quad \text{vs.} \quad H_1^{\text{non}} : \Delta > 0.$$

$$H_0 : \Delta \leq 0 \quad \text{vs.} \quad H_1 : \Delta > 0.$$

The corresponding one-sided p-value is

$$p = \frac{1 + \#\{b : \Delta^{(b)} \leq 0\}}{B + 1},$$

with rejection of H_0^{non} at level α interpreted as evidence that M_T *was not* trained on D. We use a standard value of $\alpha = 0.05$.

The addition of one to the numerator and denominator is a finite sample correction which ensures that the p-value is always greater than 0 (Phipson and Smyth, 2016; Ojala and Garriga, 2010).

5 Results

Reference models: We use the Pythia model series (Biderman et al., 2023) and OLMo-1b (Groeneveld et al., 2024) model as reference models, as the datasets used to train these models, as well as their validation datasets, are publicly available, allowing us to precisely identify data that was excluded from training. All Pythia reference models are the 'deduped' checkpoints trained on a deduplicated version of the Pile.

Datasets: We use datasets that have not previously been used to train our target model or reference model of interest, by using their validation sets. This ensures that any observed differences in rank correlation are attributable to training exposure rather than incidental overlap in pretraining corpora.

1. **The Pile**: We use the deduplicated subsets of the Pile (Gao et al., 2020) from the domains of Wikipedia (Wiki), and Pubmed Central abstracts (PubMed), ArXiV, Common Crawl (CC) released by Duan et al. (2024). These datasets were deduplicated to be especially challenging for MIA methods. We also use the Ubuntu-IRC chats, Enron emails, and Freelaw datasets that were also held out from training the Pythia models and released by Maini et al. (2024). The Pythia models are used as reference models for all Pile datasets.
2. **Dolma**: We use the Reddit held-out subset of Dolma obtained from Paloma (Soldaini et al., 2024; Magnusson et al., 2024). This was further processed to ensure that all documents had a timestamp after the cut-off date of the Pythia models to ensure they were not seen during their training. As this dataset was known to be held out from training the OLMo models, we use the OLMo-1b as the reference model for the Reddit dataset. This allows us to test the effect of having a different architecture for the target and reference model. The OLMo and Pythia model series share a similar tokenizer with the OLMo tokenizer, having some additional special tokens for masking personally identifiable information.

   See Appendix A for additional details on datasets.

We continue pretraining the Pythia-410m model, which we call Pythia-410m-CPT, to use as a target model. Each dataset from the sources above was divided into two parts, one of which was used for training and one was held out from training. In addition to the datasets above, we sample 6 billion tokens from the Common Pile dataset (Kandpal et al., 2025) for pretraining (See Appendix B). Each dataset used for training constitutes less than 0.001 % of the pretraining corpus.

5.1 Non-membership detection

From the results of Table 2, we see that PRISM is able to detect non-members for all datasets cor-

Table 2: p-values for non-member detection. None of the datasets below were used while training the vanilla Pythia models, but were used to train Pythia-410m-CPT. Pythia-410m is used as the reference model for all cases except Pythia-410m and Pythia-410m-CPT where Pythia-1b was the reference model.

Target Model	PubMed	HN	ArXiv	CC	Ubuntu	Freelaw	Enron	Reddit
Pythia-410m-CPT	1.000	0.354	0.113	0.302	0.162	0.982	0.128	0.817
Pythia-410m	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4	2.0e-4	1.0e-4
Pythia-1b	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4
Pythia-2.8b	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4
Pythia-6.9b	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4	2.0e-4	1.0e-4

Table 3: p-values computed using PRISM on Pythia-410m-CPT using a split of each of the below datasets that was held out during continued pre-training

	PubMed	HN	ArXiV	CC	Reddit	Ubuntu	Freelaw	Enron
p-value	0.84	0.01	1.0e-4	0.02	0.03	1.0e-4	1.0e-4	1.0e-4

rectly with a threshold of $p < 0.05$. When we test PRISM using Pythia-410-CPT, all p-values are well above our threshold, indicating that PRISM did not falsely rule out any datasets as being non-members. Crucially, these results also hold true for the Reddit dataset, which used a reference model (OLMo-1b) of a different architecture than the target model. The values of ρ_{RT}, ρ_{DT}, and the confidence interval for the difference in rank correlation corresponding to Table 2 are provided in Appendix H. To further evaluate the robustness of PRISM, we compute p-values using splits of each of our datasets that were held out from training Pythia-410m-CPT (Table 3). In all cases except one, PRISM is able to detect non-membership correctly. In the case of PubMed, the test is inconclusive, which is one possible outcome of the test. The lowest value observed is 10^{-4} as 10,000 bootstrap samples are drawn, and this is thus the lowest p-value that can be obtained due to the finite sample correction in the computation of the p-value. See Appendix I for additional results.

We also study the performance of existing dataset-level membership detection approaches, LLM-DI and PaCoST (Zhang et al., 2024a), at the task of dataset-level membership detection using Pythia-410m and Pythia-410m-CPT (See Appendix D for more details). When using a strict p-value threshold of 0.05, LLM-DI can detect only two out of five datasets correctly (Table 4). LLM-DI has high p-values for three datasets included in the training, indicating that the test failed to conclusively determine whether these datasets were included in the training. Thus, non-membership detection is not simply the converse of detecting membership. A high p-value is not a guarantee of non-membership. In the case of PaCoST, we find that regardless of whether the dataset was included in training, membership was either indicated for both Pythia-410m and Pythia-410m-CPT or could not be concluded for either. Thus, the test does not provide a meaningful differentiating signal. PaCoST works by computing the confidence of a given document against its paraphrases, assuming that documents included in training will have higher confidence relative to paraphrases compared to documents not included in training. As pointed out in Rastogi et al. (2025), the paraphrasing itself introduces a distribution shift, causing the test to detect lexical shifts instead of training signal.

5.2 Ablations

Ablating over the reference models. We vary the reference model used with PRISM for performing non-member detection with Pythia-410m and Pythia-410m-CPT (Table 5). Across all datasets, Pythia-1b provides the most consistent signal. Smaller models have lower representational capacity and therefore exhibit weaker alignment in rank correlations with the target model. These results highlight that stronger and more stable correlations are obtained when the reference model has sufficient scale and capacity.

Varying the size of the dataset. In Figure 3, we see that as the size of the dataset is reduced, the p-value for non-member detection stays stable until

Table 4: p-values for dataset level membership detection

Method	PubMed		HN		ArXiv		CommonCrawl		Reddit	
	410m	410m-CPT	410m	410m-CPT	410m	410m-CPT	410m	410m-CPT	410m	410m-CPT
LLM-DI	0.604	0.775	0.395	**0.022**	0.781	0.083	0.642	**0.045**	0.437	0.064
PaCoST	**3.0e-13**	**0.045**	0.683	0.999	0.158	0.122	**0.002**	**0.009**	**0.001**	**4.0e-9**

Table 5: p-values for different reference models used for non-membership detection on Pythia-410m and Pythia-410m-CPT

Reference Model	PubMed		HN		ArXiv		CommonCrawl	
	410m	410m-CPT	410m	410m-CPT	410m	410m-CPT	410m	410m-CPT
Pythia-70m	0.080	0.858	**1.0e-4**	**0.002**	**2.0e-4**	**0.030**	**0.043**	**0.030**
Pythia-160m	**1.0e-4**	0.461	**2.0e-4**	**0.017**	**1.0e-4**	**0.012**	**1.0e-4**	**4.0e-4**
Pythia-1b	**1.0e-4**	1.000	**1.0e-4**	0.354	**1.0e-4**	0.113	**1.0e-4**	0.302

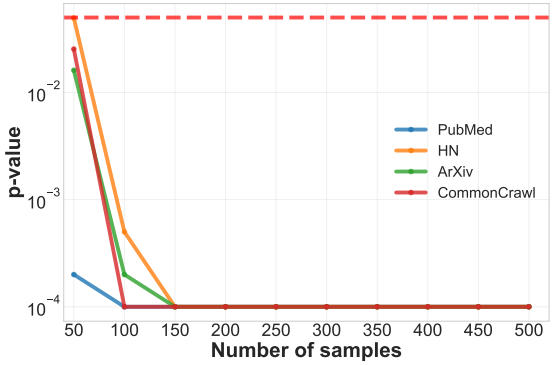
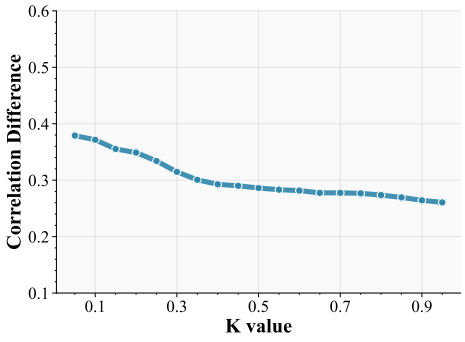
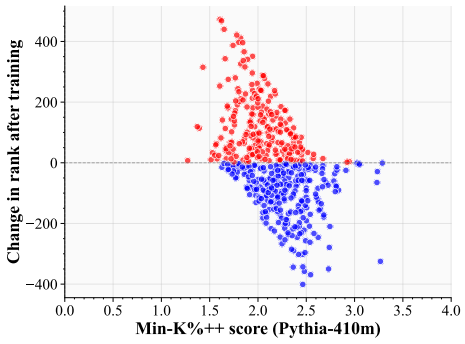


Figure 3: p-value for non-member detection when different number of samples are used



(a)



(b)

Figure 4: a) Spearman rank correlation difference of Pythia-410m and Pythia-410m-CPT against Pythia-1b reference model at different values of K in Min-K%++ b) Change in rank of a document when Min-K%++ is computed using Pythia-410m-CPT relative to the rank of the Pythia-410m model.

fewer than 150 documents are used, thus making this very practical for real-world audit scenarios.

5.3 Effect of Min-K%++

We analyze the behavior of Min-K%++ score in this section by taking the ArXiV dataset as an example. See Figure 6 and Figure 7 in the Appendix for plots corresponding to other datasets.

Changing the value of K. As the value of K increases, we observe that the correlation difference between Pythia-410m and Pythia-410m-CPT against a reference model drops. This trend indicates that the strongest training signal is concentrated in the smallest $K\%$ of each document's normalized token log-probabilities. Larger values of K include more predictable tokens, diluting the influence of the high-surprisal tokens that are most affected by memorization.

Studying change in rank with Min-K%++ value. We study the change in rank of Min-K%++ scores between Pythia-410m and Pythia-410m-CPT as a function of the Min-K%++ score of Pythia-410m (Figure 4b). The Min-K%++ score has been negated from the original definition (Zhang et al., 2025) so that higher Min-K%++ indicates higher average surprisal of the bottom K tokens for the model. Ranks are assigned in ascending order of Min-K%++ score. The results indicate that doc-

uments initially having a high Min-K%++ score drop in rank after training, while documents with low Min-K%++ scores go up in rank. This suggests that documents originally judged as "difficult" by the model become easier—showing reduced surprisal—once they are included in training. Thus, high-Min-K%++ documents exhibit the strongest memorization effect, supporting our use of Min-K%++ as a robust signal for detecting non-membership.

6 Conclusions

We presented PRISM, a novel method for showing with high confidence that a dataset has *not* been used for training. While much past work has focused on detecting members, we flip the question to ask if non-member datasets can be detected with high confidence. We find that across eight datasets, PRISM is indeed able to detect non-members while not falsely ruling out datasets that were used during training. PRISM does not require access to a reference dataset, a common requirement for many other methods, and hence makes it a practical tool for compliance audits of LLMs.

Limitations

PRISM relies on identifying when the dataset was released and for certain types of data, this may not be straightforward to establish. PRISM will be applicable for newer data after open-source LLMs became widely available. Our continued pre-training for Pythia-410m-CPT serves as an approximation to full pre-training of a model, and longer training runs may affect our results. PRISM also assumes that an entire dataset was either included or not included in training and cannot handle cases when a dataset was only partially included.

Disclaimer

References

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. 2024. Towards cross-tokenizer distillation: the universal logit distillation loss for llms. *arXiv preprint arXiv:2402.12030*.

Blake Brittain. 2024. Authors sue Anthropic for copyright infringement over AI training. *Reuters*. Technology/Artificial Intelligence section.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, and 1 others. 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650.

Bowen Chen, Namgi Han, and Yusuke Miyao. 2025a. A statistical and multi-perspective revisiting of the membership inference attack in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22854–22874, Vienna, Austria. Association for Computational Linguistics.

Yijie Chen, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025b. Enhancing cross-tokenizer knowledge distillation with contextual dynamical mapping. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8005–8018, Vienna, Austria. Association for Computational Linguistics.

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

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. *arXiv preprint arXiv:1906.01081*.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh

Hajishirzi. 2024. Do membership inference attacks work on large language models? In *Conference on Language Modeling (COLM)*.

Bradley Efron. 1992. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pages 569–593. Springer.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

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, and 24 others. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Walsh, Yanai Elazar, Kyle Lo, and 1 others. 2024. Paloma: A benchmark for evaluating language model fit. *Advances in Neural Information Processing Systems*, 37:64338–64376.

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

Markus Ojala and Gemma C Garriga. 2010. Permutation tests for studying classifier performance. *Journal of machine learning research*, 11(6).

Belinda Phipson and Gordon K Smyth. 2016. Permutation p-values should never be zero: calculating exact p-values when permutations are randomly drawn. *arXiv preprint arXiv:1603.05766*.

Haritz Puerto, Martin Gubri, Sangdoo Yun, and Seong Joon Oh. 2025. Scaling up membership inference: When and how attacks succeed on large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4165–4182, Albuquerque, New Mexico. Association for Computational Linguistics.

Saksham Rastogi, Pratyush Maini, and Danish Pruthi. 2025. Stamp your content: Proving dataset membership via watermarked rephrasings. *Preprint*, arXiv:2504.13416.

Patrick Schober, Christa Boer, and Lothar A Schwarte. 2018. Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia*, 126(5):1763–1768.

Pranav Shetty, Mirazul Haque, Petr Babkin, Zhiqiang Ma, Xiaomo Liu, and Manuela Veloso. 2025. Perturb your data: Paraphrase-guided training data watermarking. *arXiv preprint arXiv:2512.17075*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *Preprint*, arXiv:2310.16789.

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

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, and 1 others. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*.

Jonathan Stempel. 2023. NY Times sues OpenAI, Microsoft for infringing copyrighted work. *Reuters*. Legal/Transactional section.

Mohammad Tabatabai, Stephanie Bailey, Zoran Bursac, Habib Tabatabai, Derek Wilus, and Karan P Singh. 2021. An introduction to new robust linear and monotonic correlation coefficients. *BMC bioinformatics*, 22(1):170.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. *arXiv preprint arXiv:2402.09739*.

Rand Wilcox. 2017. *Modern statistics for the social and behavioral sciences: A practical introduction*. Chapman and Hall/CRC.

Rand R Wilcox. 2016. Comparing dependent robust correlations. *British Journal of Mathematical and Statistical Psychology*, 69(3):215–224.

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

Huixuan Zhang, Yun Lin, and Xiaojun Wan. 2024a. Pacost: Paired confidence significance testing for benchmark contamination detection in large language models. *arXiv preprint arXiv:2406.18326*.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2025. Min-k%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations*.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024b. Pretraining data detection for large language models: A divergence-based calibration method. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5263–5274, Miami, Florida, USA. Association for Computational Linguistics.

Bihe Zhao, Pratyush Maini, Franziska Boenisch, and Adam Dziedzic. 2025. Unlocking post-hoc dataset inference with synthetic data. In *Forty-second International Conference on Machine Learning*.

Appendix

A Datasets: Additional details

We pick the datasets from Duan et al. (2024) 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 (Puerto et al., 2025).

Reddit: We use the Reddit subset of Dolma available in the Paloma dataset (Magnusson et al., 2024). 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 1300 documents, which we truncated as above to 512 tokens with between 100 and 200 words. We further split this into 650 documents used for training and 650 documents used as held-out data. The number of documents and tokens in each dataset is shown in Table 6. All datasets and models used in our work are released under permissive open source licenses.

Table 6: Dataset statistics

Dataset	Num. Documents	Tokens
PubMed	500	154,387
HN	500	160,637
ArXiv	500	162,267
CommonCrawl	500	129,029
Reddit	650	125,789
Ubuntu	688	294,341
Freelaw	990	278,770
Enron	359	118,004

B Continued Pretraining of Pythia-410m

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 (Kandpal et al., 2025). 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 January 2021, 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) and a random seed of 1234 for all our experiments.

C Training distilled reference model

For each distilled reference model, we train on the corresponding dataset for a single epoch with a learning rate of 5×10^{-5} and a warmup of 5% of the total steps. We use a batch size of 4 and 4 gradient accumulation steps.

D Training data detection baselines

1. **LLM-DI**: LLM-DI (Maini et al., 2024) utilizes multiple MIA scores as features to train a classifier for detecting training data. To ascertain whether a model has been trained on a dataset, the verifier must generate features from both the suspect dataset and the non-member dataset and train a classifier. Binary classification using any MIA score yields an ROC-AUC score close to 0.5 (Chen et al).

Thus distinguishing any given member document from non-member document is difficult for this classifier. However, when the score of this classifier is averaged over all documents in the dataset and compared against a non-member dataset, a strong signal is produced. To calibrate this signal, LLM-DI assumes access to a non-member dataset from the same distribution. When the average classifier score of a suspect dataset is lower than that of the non-member dataset, membership can be concluded. The threshold here is determined by hypothesis testing. Given the reference assumed, the equivalent way to infer non-membership is to assume access to a member dataset from the same distribution as the dataset of interest and to flip the formulation of LLM-DI. In general, this cannot be known for an arbitrary target model. Thus, addressing this problem requires introducing a different reference against which to calibrate non-membership, which is where PRISM comes in.

In our setup, we used the splits of each dataset held out from training Pythia-410m-CPT (during continued pretraining) as the non-member documents. The implementation provided by the authors of LLM-DI was utilized in our study.

2. **PaCoST**: PaCoST (Zhang et al., 2024a) is a statistical method designed to detect QA benchmark contamination in LLMs. The approach works by first rephrasing each data input to create a counterpart with the same meaning and difficulty but different wording. Both the original and rephrased questions are then presented to the model, and the model is prompted to evaluate if the given QA pairs are correct or not based on 'Yes' or 'No' output.

   Next, the model's confidence in its answers is estimated using the output probability of the token 'Yes'. PaCoST compares the confidence scores for the original and rephrased questions using a paired samples t-test to determine if the model is significantly more confident on the original benchmark data. If the statistical test yields a p-value less than 0.05, it indicates likely contamination, meaning the model has probably seen the benchmark data during training.

PaCoST was designed to detect contamination for QA benchmarks. To use PaCoST for our datasets, we have modified the prompts for rephrasing and confidence estimation. The modified prompts are given below.

PaCoST Confidence Prompt

You are an expert in judging whether the text is correct. You will be given a text. Your job is to determine whether this text is correct. You should only respond with Yes or No. For example, given question "The value of 1+1 is 2.", the correct response should be "Yes".

PaCoST Rephrasing Prompt

You are provided with a text. Your task is to rephrase this text into another text with the same meaning. When rephrasing the text, you must ensure that you follow the following rules: (1). You must ensure that you generate a rephrased text as your response. (2). You must ensure that the rephrased text bears the same meaning with the original text. Do not miss any information. (3). You must only generate a rephrased text. Any other information should not appear in your response. (4). Do not output any explanation. (5). Do not modify the numbers or quantities in the question. You should remain them unchanged. Example 1: given text "1+1=2", one possible response should be: The value of 1+1 is 2. Example 2: given text "Earth orbits around the Sun", one possible response should be: The Sun is the center of earthś orbit. Example 3: given text "C is the third letter in English", one possible response should be: Englishś third letter is C. Example 4: given text "A scientist is looking for something to watch faraway stars.", one possible response should be: A scientist would like use something to watch remote stars. Example 5: given text "John has 8 cats and five dogs. Linda has 6 rabbits.", one possible response should be: John owns 8 cats and five dogs. Linda possesses 6 rabbits.

E Membership Inference

Below, we summarize some key methods used in the literature.

- **Perplexity/Loss** (Yeom et al., 2018). Uses the model's loss (perplexity for language models) as the score: lower loss implies higher likelihood of membership.
- **Min-K%** (Shi et al., 2023). Averages the probabilities of the *least likely* $K\%$ tokens in a sequence.
- **Zlib Entropy** (Carlini et al., 2021). Computes a score as the ratio between model perplexity and the `zlib` compression size of the text. Smaller ratios indicate potential membership.
- **DC-PPD** (Zhang et al., 2024b). Detects membership by measuring the divergence between the model's token-probability distribution and the empirical token-frequency distribution of the text.

F Correlation trends

In Table 7, we compare the correlation difference using well-known MIA scores in addition to Min-K%++ to justify the choice of using Min-K%++. The ranking is computed using each of the scores in the table for each dataset, where the reference model for computing correlations against was Pythia-1b. Observe that Min-K%++ was the score that consistently gave the largest differences, thus making it a suitable choice as a proxy for the training signal.

G Effect of Distillation Hyperparameters

We study the effect of the distillation hyperparameters—the *temperature* (τ) and the *distillation weight* (λ)—on the correlation difference using the Wikipedia dataset (Figure 5). The temperature τ controls the smoothness of the target model's probability distribution during distillation, while λ balances the contribution of the knowledge-distillation loss and the cross-entropy loss on the suspect dataset. The goal is to pick λ and τ such that the correlation difference is positive for Pythia-410m (detecting true positives) and negative for Pythia 410m-CPT to avoid the possibility of false positives.

We first vary τ over a grid while fixing $\lambda = 0.5$. We identify $\tau = 2$ as the optimal value, as it yields

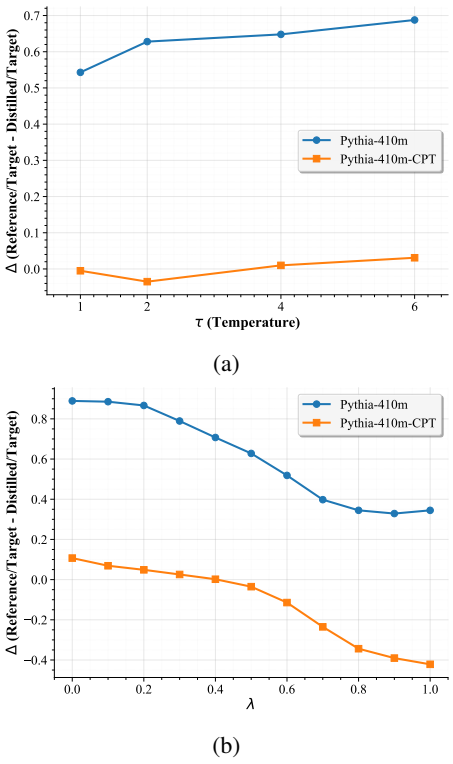


Figure 5: Hyperparameter study to determine the value of a) τ and b) λ. Pythia-410m and Pythia-410m-CPT are the target models while Pythia-1b is the reference model.

a negative correlation difference for Pythia-410m-CPT, allowing members to be ruled out with high confidence. With τ fixed at this value, we next vary λ between 0 and 1. We observe that the correlation difference decreases as λ increases for both Pythia-410m and Pythia-410m-CPT. We pick the smallest λ that prevents false positives. At $\lambda = 0.6$ the correlation difference is -0.1, which still leaves open the possibility of false positives, hence we pick $\lambda = 0.7$ where the correlation difference is -0.25. We do not pick a higher λ because that would lead to lower correlation difference for detecting true positives. Accordingly, we select $\lambda = 0.7$ and $\tau = 2$ as our default hyperparameters. These hyperparameters were applied to all datasets and target models of different sizes in Table 2.

The sensitivity to hyperparameters can also be observed from Figure 5. Detecting true positives works at all hyperparameter values, as the correlation difference is always much greater than 0 for Pythia-410m. To avoid false positives, however, it is necessary to pick $\lambda > 0.6$.

Table 7: Spearman rank correlation of Pythia-410m and Pythia-410m-CPT models against Pythia-1b reference model computed using different MIA scores

	PubMed			HN			ArXiv			CommonCrawl			Wiki		
Score	410m	CPT	Δ	410m	CPT	Δ	410m	CPT	Δ	410m	CPT	Δ	410m	CPT	Δ
Loss	0.99	0.92	0.07	0.98	0.83	0.15	0.99	0.84	0.15	0.98	0.72	0.26	0.98	0.55	0.43
zlib	0.99	0.93	0.06	0.99	0.93	0.06	0.99	0.89	0.10	0.98	0.82	0.16	0.98	0.57	0.41
DC-PDD	0.98	0.92	0.06	0.95	0.62	0.33	0.97	0.85	0.12	0.90	0.50	**0.40**	0.98	0.47	0.51
Min-K%	0.96	0.79	0.17	0.90	0.64	0.26	0.93	0.59	0.34	0.94	0.75	0.19	0.94	0.42	0.52
Min-K%++	0.84	0.33	**0.51**	0.75	0.41	**0.34**	0.82	0.45	**0.37**	0.79	0.51	0.28	0.84	0.25	**0.59**

H Rank correlation results

Tables 8 and 9 show the raw rank correlations and the corresponding confidence intervals for the p-values in Table 2.

I Additional results

I.1 Additional baseline

While PRISM uses rank correlations, we test whether directly using the Min-K%++ scores can be used to infer non-membership. We use the Min-K%++ scores from the target model and the distilled reference model. As the distilled reference model is trained on the data set, its Min-K%++ scores are expected to be higher than those of the target model if the target model has not been trained on the data set. Formally, the null hypothesis is that the Min-K%++ scores under both models are equal and can be written as:

$$H_0 : \mathbb{E}_x[f_{\text{Min-K\%++}}(x; M_T)] = \mathbb{E}_x[f_{\text{Min-K\%++}}(x; M_D)] \quad (1)$$

while the alternate hypothesis can be written as:

$$H_1 : \mathbb{E}_x[f_{\text{Min-K\%++}}(x; M_T)] < \mathbb{E}_x[f_{\text{Min-K\%++}}(x; M_D)] \quad (2)$$

Note that establishing non-membership requires comparing against a model where the dataset is known to be a member, and thus, to establish non-membership of $\mathcal{D}$ in M_T requires using M_D as the reference. To compute a p-value, we perform a paired t-test over each document in the data set.

The results (Table 10) show that non-member detection is inconsistent, as evidenced by HackerNews and Enron for Pythia-2.b and Pythia-6.9b, where non-membership fails to be detected. For Pythia-410m-CPT, in all cases, non-membership is falsely concluded. This shows that using Min-K%++ scores alone is not a reliable way to infer non-membership. As Min-K%++ values from different models are being compared, the difference in Min-K%++ values between them would arise not only due to a difference in membership of the dataset in training but also due to differences in the number of parameters and the architectures of the models.

I.2 OLMo-1b as the target model

We test PRISM using OLMo-1b as the target model on the Reddit dataset with various Pythia models as the reference models (Table 11). In all cases, non-membership was successfully detected. The Reddit dataset is non-member for all the Pythia models, as all documents in it have a release date after the knowledge cut-off of the Pythia models.

I.3 Rank correlation metric

While we have used the Spearman correlation metric for computing rank correlations everywhere in our work, we find that computing the rank correlation with Kendall τ would also work equally well and gives similar results for non-membership detection (Table 12).

Table 8: Spearman rank correlations using for each target model and dataset. This is the raw data corresponding to the first four datasets in Table 2. ρ_{RT} is the Spearman rank correlation between the Min-K%++ scores from the reference and target model over the dataset, ρ_{DT} is the Spearman rank correlation between the Min-K%++ scores from the distilled reference and target model and $\Delta_{95\%CI}$ is the 95 % confidence interval for $\rho_{RT} - \rho_{DT}$ obtained by bootstrapping.

Target Model	PubMed			HN			ArXiv			CommonCrawl		
	ρ_{RT}	ρ_{DT}	$\Delta_{95\%CI}$	ρ_{RT}	ρ_{DT}	$\Delta_{95\%CI}$	ρ_{RT}	ρ_{DT}	$\Delta_{95\%CI}$	ρ_{RT}	ρ_{DT}	$\delta_{95\%CI}$
Pythia-410m-CPT	0.328	0.590	[-0.371, -0.151]	0.404	0.384	[-0.080, 0.121]	0.447	0.393	[-0.034, 0.142]	0.512	0.488	[-0.068, 0.116]
Pythia-410m	0.836	0.420	[0.322, 0.511]	0.752	0.467	[0.207, 0.366]	0.819	0.506	[0.232, 0.394]	0.791	0.446	[0.261, 0.434]
Pythia-1b	0.844	0.513	[0.249, 0.416]	0.722	0.512	[0.127, 0.291]	0.812	0.539	[0.200, 0.348]	0.840	0.482	[0.282, 0.439]
Pythia-2.8b	0.783	0.498	[0.203, 0.369]	0.661	0.307	[0.262, 0.446]	0.760	0.443	[0.239, 0.396]	0.807	0.361	[0.356, 0.536]
Pythia-6.9b	0.757	0.385	[0.283, 0.463]	0.651	0.321	[0.240, 0.420]	0.733	0.452	[0.204, 0.358]	0.769	0.302	[0.371, 0.563]

Table 9: Spearman rank correlations using for each target model and dataset. This is the raw data corresponding to the last four datasets in Table 2. ρ_{RT} is the Spearman rank correlation between the Min-K%++ scores from the reference and target model over the dataset, ρ_{DT} is the Spearman rank correlation between the Min-K%++ scores from the distilled reference and target model and $\Delta_{95\%CI}$ is the 95 % confidence interval for $\rho_{RT} - \rho_{DT}$ obtained by bootstrapping.

Target Model	Ubuntu			Freelaw			Enron			Reddit		
	ρ_{RT}	ρ_{DT}	$\Delta_{95\%CI}$	ρ_{RT}	ρ_{DT}	$\Delta_{95\%CI}$	ρ_{RT}	ρ_{DT}	$\Delta_{95\%CI}$	ρ_{RT}	ρ_{DT}	$\delta_{95\%CI}$
Pythia-410m-CPT	0.312	0.259	[-0.054, 0.160]	0.411	0.490	[-0.151, -0.006]	0.432	0.355	[-0.055, 0.208]	0.449	0.488	[-0.123, 0.045]
Pythia-410m	0.784	0.266	[0.429, 0.605]	0.826	0.591	[0.184, 0.289]	0.703	0.440	[0.137, 0.392]	0.672	0.080	[0.503, 0.678]
Pythia-1b	0.767	0.395	[0.302, 0.441]	0.825	0.498	[0.277, 0.377]	0.646	0.287	[0.226, 0.496]	0.721	0.070	[0.560, 0.736]
Pythia-2.8b	0.718	0.308	[0.331, 0.489]	0.753	0.398	[0.298, 0.413]	0.530	0.188	[0.202, 0.479]	0.763	0.052	[0.626, 0.793]
Pythia-6.9b	0.675	0.298	[0.295, 0.459]	0.708	0.359	[0.289, 0.410]	0.443	0.178	[0.126, 0.405]	0.774	0.038	[0.650, 0.817]

Table 10: Non-Member detection baseline results using Min-K%++ values from target and distilled reference model.

Target Model	PubMed	HN	ArXiv	CommonCrawl	Ubuntu	Freelaw	Enron	Reddit
Pythia-410m-CPT	2.6e-5	1.7e-24	3.7e-45	8.0e-8	1.2e-16	9.9e-34	3.6e-8	2.5e-8
Pythia-410m	0.18	9.6e-22	9.5e-11	2.8e-21	0.54	0.03	5.7e-17	2.5e-6
Pythia-1b	1.9e-11	2.2e-3	2.6e-47	7.0e-10	1.2e-18	3.7e-35	0.84	3.3e-6
Pythia-2.8b	7.8e-15	0.63	2.3e-8	0.06	0.04	1.6e-31	0.36	2.4e-6
Pythia-6.9b	4.8e-5	1.0	2.8e-38	0.01	2.7e-4	0.03	0.92	2.0e-6

Table 11: p-values for different Pythia reference models with OLMo-1b as the target model for the Reddit dataset.

	Pythia-70m	Pythia-160m	Pythia-410m	Pythia-1b
p-value	1.0e-4	1.0e-4	1.0e-4	1.0e-4

Table 12: Non-Member detection p-values using PRISM where the rank correlation is computed using Spearman ρ and Kendall τ

Correlation Method	PubMed	HN	ArXiv	CommonCrawl	Reddit
Spearman ρ	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4
Kendall τ	1.0e-4	1.0e-4	1.0e-4	1.0e-4	1.0e-4

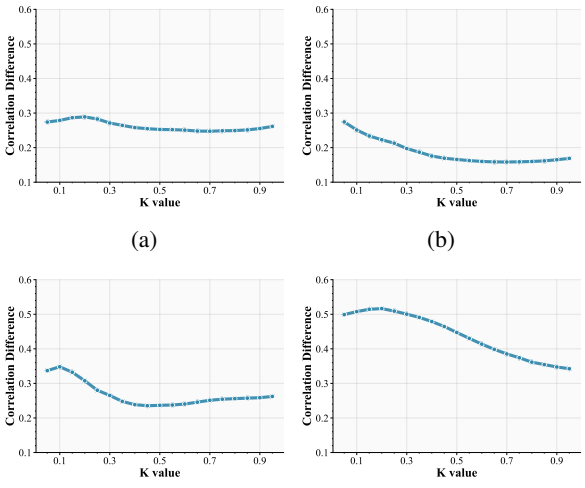


Figure 6: Spearman rank correlation difference of Pythia-410m and Pythia-410m-CPT against Pythia-1b reference model at different values of K in Min-K%++ for different datasets a) CommonCrawl b) Reddit c) HackerNews d) PubMed

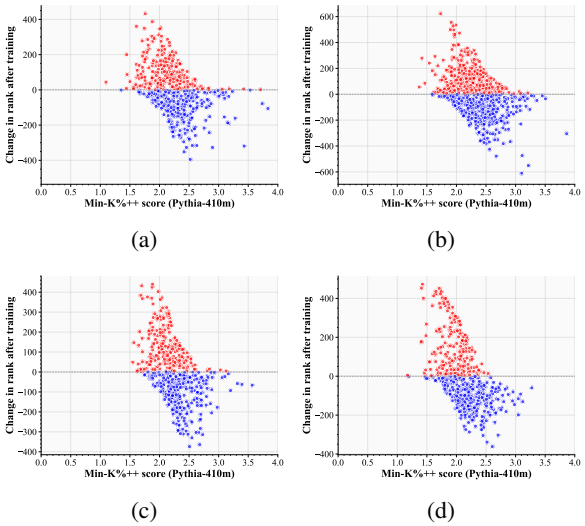


Figure 7: Change in rank of a document when Min-K%++ is computed using Pythia-410m-CPT relative to the rank of the Pythia-410m model for different datasets a) CommonCrawl b) Reddit c) HackerNews d) PubMed