Hey, That's My Data! Token-Only Dataset Inference in Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) rely on massive training datasets, often including proprietary data, which raises concerns about unauthorized usage and copyright infringement. Existing dataset inference methods typically require access to log probabilities or other internal signals, but many modern LLMs restrict such access, motivating token-only inference approaches. We propose CatShift, a token-only dataset inference framework based on catastrophic forgetting, where models overwrite prior knowledge when trained on new data. Fine-tuning an LLM on a subset of its training data induces larger output shifts than fine-tuning on unseen data. CatShift compares these shifts against those from a known non-member validation set to infer whether a dataset was included in training. Experiments on both open-source and API-based LLMs show that CatShift remains effective without logit access, enabling practical protection of proprietary datasets.

1 Introduction

Large Language Models (LLMs) are increasingly trained on massive web-scale corpora, enabling strong performance in language understanding, reasoning, and generation. However, reliance on large and opaque datasets raises a fundamental question: was a specific dataset, potentially copyrighted or proprietary, used during training? Answering this question is crucial for copyright protection, licensing compliance, and accountability in modern LLM deployment. Concerns over unauthorized data use have intensified with the commercialization of LLMs. Copyrighted material may be included in training data without consent, infringing authors' rights and causing financial harm (Wei et al., 2024; Ren et al., 2024; Liu et al., 2024). A prominent example is the lawsuit filed by the New York Times against OpenAI and Microsoft over alleged misuse of its content (The New York Times, 2023). Media organizations invest heavily in curating high-quality datasets and rarely release them publicly, while even open corpora (Raffel et al., 2020; Gao et al., 2021) often restrict usage to research or educational purposes. Detecting whether such datasets were used for training is essential.

Challenges in Dataset Inference. Dataset inference (Maini et al., 2024, 2021; Dziedzic et al., 2022) seeks to determine whether a model was trained on a given dataset, typically by analyzing logits or log-probabilities. While effective in white-box or logit-accessible settings, these methods fail when models expose only top-1 predictions, a restriction already enforced by commercial systems such as Claude (Anthropic, 2024). This limitation motivates *token-only* inference, which relies solely on predicted tokens. Although token-only access is widely available and robust to logit-perturbation defenses (Fang et al., 2024), naïve strategies such as equating membership with good completion accuracy are unreliable, as shown in Section 4.2. More principled approaches are needed.

Our Solution. We observe that fine-tuning can be repurposed not only for adaptation, but also to probe what models implicitly remember. Our approach leverages *catastrophic forgetting* (CF) (Luo et al., 2023), where learning new data overwrites prior knowledge. If a dataset was part of pre-training, fine-tuning on it again causes previously learned representations to resurface, leading to a pronounced shift in outputs. In contrast, fine-tuning on unseen data induces less structured changes.

CatShift operationalizes this insight using fine-tuning APIs supported by major providers (OpenAI, Hugging Face, Google Vertex AI, Claude). The candidate dataset is split into train and test subsets. The model is fine-tuned on the train portion, and output shifts are measured on the test portion. Larger shifts indicate prior exposure, while a held-out validation set guaranteed to be unseen provides

a non-member baseline. To quantify this effect, we compute similarity scores between pre- and post-tuning outputs (e.g., using a BERT-based scorer) and compare them with validation-set scores. Statistical tests such as KS or Mann–Whitney U assess whether the observed shift is significantly larger for the target dataset. A low p-value indicates likely membership.

Results. Our method achieves strong and consistent performance across both open-source and API-based language models, with substantial improvements over prior dataset inference baselines. On the Pythia family of models (Biderman et al., 2023), CatShift achieves an AUC of 0.979 on Pythia-410M, outperforming the state-of-the-art methods: PDD (Zhang et al., 2024) by more than 0.49 AUC points and LLM DI (Maini et al., 2024) by 0.19 AUC points. At a stringent operating point of 5% false positive rate, CatShift further maintains high true positive rates, whereas both baselines degrade sharply under token-only access, with PDD exhibiting near-random behavior. On GPT-3.5 with the BookMIA dataset (Shi et al., 2024), member books yield $p = 6.44 \times 10^{-5}$ while non-members yield $p = 0.711$, demonstrating robust discrimination under token-only access. Overall, these results show that dataset inference remains feasible even when models expose only top-1 predictions, enabling practical copyright protection without access to logits or privileged internal signals.

Contributions.

- We present the first systematic study of dataset inference in token-only settings, targeting realistic scenarios where log probabilities are unavailable.
- We introduce CatShift, a CF-based inference approach that consistently outperforms existing baselines across models and datasets.
- We provide empirical evidence that training members exhibit significantly larger output shifts after fine-tuning than non-members, explaining the effectiveness of CF-based inference.

2 Background

2.1 Dataset Inference Attack

Dataset inference attacks aim to distinguish data drawn from the model's training distribution $p_{\mathcal{D}_{\text{train}}}$ from data sampled from a non-member distribution $p_{\mathcal{D}_{\text{non-train}}}$. Following Maini et al. (Maini et al., 2021), who observed that training data often lies farther from decision boundaries, these attacks exploit distributional shifts in features, logits, or model outputs to infer membership collectively.

Formally, a scoring function $s(x)$ estimates the distance or likelihood of a sample x under $p_{\mathcal{D}_{\text{train}}}$. A threshold τ is applied: $s(x) > \tau \Rightarrow x \in \mathcal{D}_{\text{train}}$; otherwise, $x \in \mathcal{D}_{\text{non-train}}$. Recent work (Maini et al., 2024) extends this to LLMs, where fine-tuning induces measurable distribution shifts. Hypothesis testing can then be used for membership inference, but requires labeled member/non-member data and a separate validation set for calibrating τ.

2.2 Fine-Tuning APIs

Fine-tuning APIs are now widely offered for LLMs, enabling users to adapt powerful pre-trained models to specific domains without full retraining. Platforms like OpenAI's GPT-3.5 and Hugging Face allow developers to fine-tune models on custom data for applications such as chatbots, legal document analysis, and clinical text classification.

Fine-tuning integrates domain-specific knowledge into general-purpose models, improving performance on specialized terminology and tasks. For instance, healthcare models better extract clinical entities, and legal models handle contracts more accurately. Fine-tuning also supports domain adaptation across languages and regions, boosting robustness where generic models underperform. These APIs provide model customization without exposing internals, lowering technical barriers and enabling adoption in privacy-sensitive industries. Standardized interfaces make advanced LLM capabilities accessible to a wide range of sectors.

3 Methodology of CatShift

3.1 Threat Model

Adversary and Goals. We consider an adversarial LLM provider who may have trained on proprietary data without consent. The *dataset owner* seeks to determine whether their data was used in training under restrictive query conditions, even if the provider withholds log probabilities or manipulates outputs to conceal unauthorized usage.

Defender Capabilities. The dataset owner has black-box access to the target LLM through a token-only interface, observing only *top-1 completions*, which we explicitly define as deterministic greedy decoding with temperature set to zero (i.e., the model always outputs the most likely next token at each step). No logits, confidence scores, or token probabilities are exposed. The defender can fine-tune the LLM on custom datasets via publicly

available APIs and possesses a validation set guaranteed not to be included in the model's training corpus. This non-member set serves as a baseline for measuring output distribution shifts induced by fine-tuning on candidate datasets.

Adversarial Constraints. The provider may hide membership signals by withholding log probabilities, randomizing outputs, or limiting fine-tuning options such as training duration, capacity, or number of experiments through pricing. These measures aim to suppress overfitting indicators and reduce the owner's ability to probe the model.

Our Assumptions. We assume that fine-tuning genuinely updates model parameters, that queries before and after fine-tuning yield outputs consistent with the model's current state, and that the validation set is truly unseen by the model.

Attack Surface and Desired Protections. Given these conditions, the dataset owner's strategy is to detect unauthorized data use through token-only queries and statistical comparison of fine-tuning effects on candidate versus validation datasets. This enables robust dataset inference even under adversarial interference and minimal signal exposure.

3.2 Design Motivation

LLMs are increasingly deployed across diverse applications and often trained on massive, sometimes proprietary datasets. Determining whether a specific dataset was used for training is critical for copyright protection and legal compliance. However, most existing membership-inference methods rely on outputs such as log probabilities or confidence scores—signals that many providers now suppress. This motivates the need for *token-only* approaches that rely solely on top-1 predictions or completions.

The challenge lies in the limited information available: without token likelihoods or internal activations, identifying training data is far more difficult. Our approach leverages *catastrophic forgetting*—the tendency of models to overwrite but also "reactivate" previously learned knowledge when re-exposed to training data.

Specifically, as shown in Appendix A, increasing the number of fine-tuning iterations leads to a monotonic growth in the L2 distance between the fine-tuned and original model weights, a hallmark of progressive parameter drift associated with catastrophic forgetting. At the same time, statistical tests on top-1 completions reveal increasingly significant distributional shifts when the fine-tuning data overlaps with the pre-training corpus. This inverse relationship between parameter distance and hypothesis-test p-values provides concrete evidence that forgetting during adaptation can reactivate latent knowledge tied to member datasets.

Building on this observation, we measure the *shift* in output distributions after fine-tuning on a subset of the target dataset. If the dataset was part of training, the model exhibits a pronounced shift, reflecting knowledge recovery; unseen data induces smaller shifts, indicating new learning. Comparing these shifts enables us to distinguish member from non-member data.

Because our method operates entirely on token-only outputs, it remains effective even when log probabilities are hidden. This makes it a robust, practical approach for dataset inference under modern LLM service constraints, helping safeguard the intellectual property of dataset owners without requiring access to model internals.

3.3 Overview of CatShift

To address the token-only constraint, CatShift combines *catastrophic forgetting* with *statistical hypothesis testing* to distinguish a target dataset from a known non-member validation set, without relying on confidence scores or internal states. The key insight is that fine-tuning affects member and non-member data asymmetrically at the output level, even as model parameters drift away from their pre-trained configuration.

Fine-tuning induces parameter drift characteristic of catastrophic forgetting. For member data, this process re-optimizes around latent representations already encoded during pre-training, leading to comparatively rigid or structured shifts in token-level outputs. In contrast, non-member data requires forming new representations, resulting in more diffuse and less stable output changes. This asymmetry yields a detectable statistical signal under token-only observation, as shown in Fig. 1.

Guided by this observation, CatShift measures output distribution shifts induced by controlled fine-tuning and compares them against those observed on a held-out non-member dataset.

Step 1: Completion Prompt Construction. We frame dataset inference as a text-completion task. Each sample is split into a prompt and completion, and the prompt is placed inside a generic instruction (e.g., "Complete the following text"). This

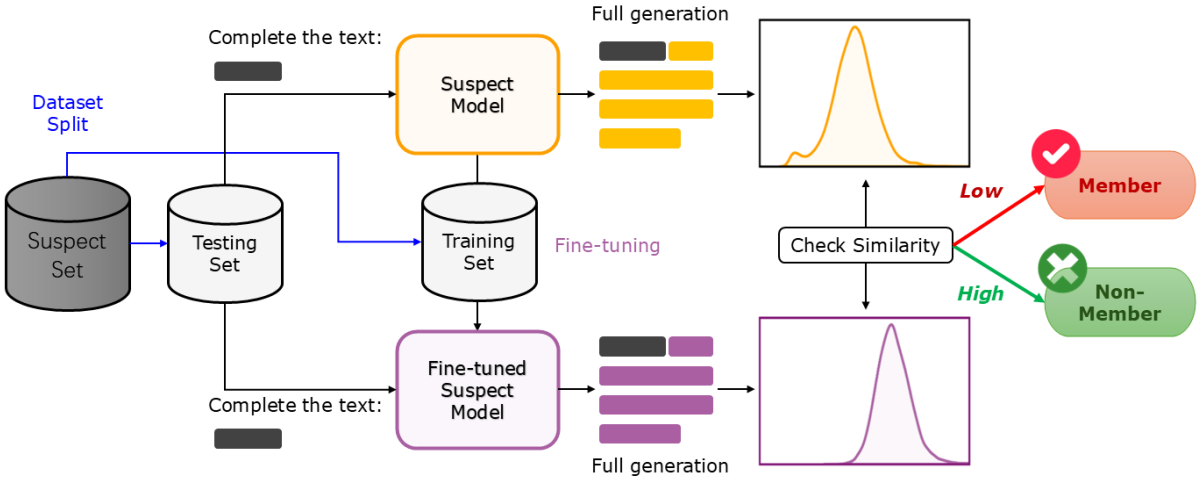

Figure 1: An overview of the CatShift framework for dataset inference on token-only large language models (LLMs). The figure illustrates the three-step process: (1) Dataset Split—where the target dataset is partitioned into training and test subsets; (2) Completion Prompt Construction—where data samples are transformed into prompt-completion pairs, ensuring compatibility with token-only models; (3) Target Model Fine-Tuning—where the LLM is fine-tuned on the training subset of the target dataset to detect shifts caused by re-exposing the model to previously seen data; (4) Output Distribution Analysis—where the model's completions before and after fine-tuning are compared using similarity metrics, followed by statistical hypothesis testing to determine whether the target dataset was part of the model's original training data.

setup works with token-only interfaces since it requires only top-1 predictions, aligning with typical commercial LLM APIs.

Step 2: Target Model Fine-Tuning. We fine-tune the target LLM on these prompt–completion pairs. If the dataset was part of the training corpus, fine-tuning rapidly refreshes internal representations, causing a pronounced shift in outputs. For unseen data, the model learns new patterns more gradually, producing smaller shifts.

Step 3: Output Distribution Analysis. We collect pre- and post-fine-tuning completions, compute similarity metrics (e.g., lexical overlap or BERT-based scores), and compare them with those from the validation set. Statistical tests determine whether the target dataset's shift is significantly larger, signaling likely membership. Because this process relies only on top-1 predictions, it remains robust even when logits or probabilities are hidden.

3.4 Completion Prompt Construction

We represent each sample in the *target dataset* as a prompt–completion pair:

$$D_s = (x_i, y_i)_{i=1}^{N}, \tag{1}$$

where x_i is the prompt and y_i the completion. In practice, we split each document in half, using the first half as the prompt and the second as the completion, ensuring consistency across samples. Under the token-only setting, the LLM produces a continuation $\hat{y}_i$ for each x_i, which we record for later analysis. This construction is fully compatible with standard LLM APIs and enables dataset owners to test for unauthorized data usage.

If the model is instruction-tuned, we directly use prompts to elicit completions, leveraging its ability to follow explicit instructions for more reliable outputs. Otherwise, we use simple prefixes to trigger causal continuations. Since models tend to assign higher likelihood to training data, prefixes from member data often lead the model to reproduce it, making prefix-based continuation a practical strategy for eliciting memorized content in non-instruction-tuned models.

3.5 Target Model Fine-Tuning

We fine-tune the target LLM $f^{(0)}$ with parameters $\theta^{(0)}$ on a subset of the target dataset. The dataset is split into D_s^{train} and D_s^{test} with no overlap, and fine-tuning updates the parameters to $\theta^{(1)}$ by minimizing

$$\theta^{(1)} = \arg\min_{\theta} \sum_{(x_i, y_i) \in D_s^{\text{train}}} \mathcal{L}\big(f^{(0)}(x_i; \theta), y_i\big),$$

where $\mathcal{L}$ is typically cross-entropy loss. If D_s was part of the original training data, re-exposure refreshes its patterns and produces a stronger change in predictions; if it is novel, the model must learn new patterns, resulting in more gradual shifts. This distinction underpins our later statistical test.

To perform fine-tuning efficiently, we use Low-Rank Adaptation (LoRA) (Hu et al., 2022), which updates a low-rank decomposition of weight matrices, reducing computational cost while preserving performance. We apply LoRA to the Query-Key-Value (QKV) attention layer, the linear layer following attention, and the MLP layer. Fine-tuning these components allows the model to adjust its attention weights, refine internal representations, and update prediction mappings, all while retaining most pretrained parameters.

By focusing adaptation on these layers, we achieve rapid, parameter-efficient fine-tuning that highlights differences between reactivating existing knowledge and learning from scratch. Member datasets trigger larger shifts in top-1 predictions, whereas non-member datasets produce subtler changes—providing a measurable signal for dataset membership inference.

3.6 Output Distribution Analysis

We quantify the *shift* in outputs by comparing completions before and after fine-tuning. For each $(x_i, y_i) \in D_s^{\text{test}}$, we collect

$$\hat{y}_i^{(0)} = f^{(0)}(x_i), \qquad \hat{y}_i^{(1)} = f^{(1)}(x_i),$$

where $\hat{y}_i^{(0)}$ and $\hat{y}_i^{(1)}$ are top-1 completions from the original and fine-tuned models.

Similarity Scores. We measure the change with a similarity function $\mathrm{Sim}(\cdot, \cdot)$ (e.g., n-gram overlap or BERT score):

$$s_i = \mathrm{Sim}\big(\hat{y}_i^{(0)}, \hat{y}_i^{(1)}\big),$$

where smaller s_i indicates greater divergence and thus a larger shift.

Validation Baseline. To calibrate what constitutes a "normal" shift, we repeat the same process on a known non-member validation set $D_v = \{(x_j^v, y_j^v)\}_{j=1}^M$, obtaining

$$s_j^v = \mathrm{Sim}\big(f^{(0)}(x_j^v),\ f^{(1)}(x_j^v)\big)$$

This yields a baseline distribution of shifts for non-member data.

Hypothesis Testing. Let $S_s = \{s_i\}$ and $S_v = \{s_j^v\}$ be the similarity score sets for the target and validation data. We apply a two-sample test such as Kolmogorov–Smirnov:

$$p\text{-value} = \mathrm{KS}(S_s, S_v).$$

If p-value $< \alpha$ (e.g., $\alpha = 0.1$), we conclude D_s exhibits a significantly larger shift and is likely a training member; otherwise, we find no strong evidence of membership.

By relying solely on token-only completions and a robust validation baseline, CatShift detects unauthorized data usage even when log probabilities are withheld or obfuscated, making it practical for real-world LLMs.

Overall Algorithm. The full pseudocode of the proposed CatShift framework is provided in Appendix B for completeness. Here, we focus on the intuition and main components that drive the framework's design.

4 Evaluation

4.1 Experimental Setup

Datasets and Models. We evaluate on the full PILE dataset (Gao et al., 2021), which comprises 22 subsets, using both public checkpoints and API-based models. For public models, we use Pythia (Biderman et al., 2023) and GPT-Neo (Black et al., 2021), both trained on PILE. Because the training and validation splits of all 22 subsets are public, we can precisely label members (training samples) and non-members (validation samples) under IID conditions with no temporal shift, using each subset's validation split as its baseline. For API-based evaluation, we fine-tune GPT-3.5 through OpenAI's API and use the BookMIA dataset (Shi et al., 2024), which contains known member and non-member books for OpenAI models. Unless stated otherwise, we sample 1000 training and 1000 validation examples per PILE subset for membership evaluation, and use an additional 600 samples for fine-tuning in CatShift.

We adhered to established ethical guidelines. Most experiments used publicly available datasets in a controlled local environment. For GPT-3.5, we fine-tuned using only public data and did not release the resulting model, mitigating misuse risks. As no human participants were involved, our IRB determined this work does not constitute human-subjects research. While misuse is possible, for example if malicious actors use dataset-inference techniques to probe private models, we disclose these findings to raise awareness and encourage constructive discussion.

Note that we used LLMs only to polish the writing style (e.g., grammar and spelling).

Baselines. We compare CatShift against two representative dataset inference baselines for large language models: LLM DI (Maini et al., 2024) and PDD (Zhang et al., 2024).

LLM DI (Maini et al., 2024) formulates dataset inference as a statistical testing problem based on differences in model loss or likelihood statistics between candidate and reference datasets. While effective when logits or loss values are accessible, it fundamentally relies on probabilistic outputs, which are unavailable under token-only interfaces.

PDD (Zhang et al., 2024) detects pretraining data by measuring calibrated divergences between output distributions induced by controlled perturbations. This approach similarly assumes access to probability distributions or calibrated likelihood estimates to compute divergence scores, limiting its applicability in strict black-box settings where only top-1 completions are exposed.

In contrast, CatShift operates solely on top-1 completions and does not require access to logits, probabilities, or loss values, making it compatible with LLM APIs that restrict output visibility.

Evaluation Protocol. We evaluate CatShift on each of the 22 PILE subsets, treating training splits as members and validation splits as non-members. For each subset, we apply the Kolmogorov–Smirnov (KS) test to compare the distributions of similarity scores obtained from candidate data and a non-member validation set:

$$D = \sup_x |F_n(x) - F_m(x)|.$$

The resulting p-values serve as continuous membership scores, with smaller values indicating stronger evidence of membership. Rather than relying on a single decision threshold, we evaluate performance using threshold-independent metrics, reporting ROC AUC to measure ranking quality and TPR at a fixed 5% false positive rate (TPR@5%) to assess sensitivity under conservative operating conditions relevant for practical auditing.

CatShift Configuration. We fine-tune with 600 samples per subset. For GPT-3.5, we adopt OpenAI's default configuration; for Pythia and GPT-Neo, we use LoRA (Hu et al., 2022) with rank 8, $\alpha = 32$, and dropout 0.1, training with learning rate 8×10^{-5} and batch size 8. Checkpoints are saved every 10 steps, and the model with the lowest normalized combined loss (on both member and non-member sets) is selected. Fine-tuning and test sets are strictly disjoint.

Table 1: Dataset inference performance across models under token-only access. We compare CatShift against two representative baselines, PDD and LLM DI, using AUC and TPR at 5% false positive rate (TPR@5%). Results show that CatShift consistently outperforms both baselines across all model scales, achieving near-perfect discrimination even when logits and probability scores are unavailable.

Model	Method	AUC	TPR@5%
Pythia 410M	PDD (Zhang et al., 2024)	0.4865	0.0499
	LLM DI (Maini et al., 2024)	0.7890	0.5910
	CatShift (Ours)	**0.9790**	**0.9545**
GPT-Neo 125M	PDD (Zhang et al., 2024)	0.5000	0.0680
	LLM DI (Maini et al., 2024)	0.9122	0.6860
	CatShift (Ours)	**0.9940**	**0.9561**
GPT-Neo 1.3B	PDD (Zhang et al., 2024)	0.4896	0.0507
	LLM DI (Maini et al., 2024)	0.8200	0.4113
	CatShift (Ours)	**0.9410**	**0.8227**
GPT-NeoX 20B	PDD (Zhang et al., 2024)	0.5574	0.0731
	LLM DI (Maini et al., 2024)	0.9280	0.7727
	CatShift (Ours)	**1.0000**	**0.9640**

Computational Cost. Although our method relies on fine-tuning, its computational cost remains modest in practice. Dataset inference requires only a small number of fine-tuning iterations on a limited subset of samples, rather than full retraining. In our experiments, performing detection with GPT-3.5 incurs an estimated cost of approximately $156 per target dataset, well within the budget of routine auditing or compliance checks. This demonstrates that fine-tuning-based dataset inference is not prohibitively expensive, and can be realistically deployed for practical ownership verification and misuse detection.

4.2 Effectiveness of CatShift

We evaluate the effectiveness of CatShift under token-only access across multiple model architectures and scales, comparing it against two representative dataset inference baselines: PDD and LLM DI. Both baselines are evaluated under their applicable access assumptions, while CatShift relies exclusively on top-1 completions.

Table 1 reports AUC and TPR at a fixed 5% false positive rate (TPR@5%), a stringent operating point relevant for practical auditing scenarios. Across all models, CatShift consistently achieves the highest AUC, indicating superior ranking quality between member and non-member datasets independent of any decision threshold. On Pythia-410M, CatShift attains an AUC of 0.979, substantially outperforming LLM DI (0.789) and PDD (0.487). Similar trends hold for GPT-Neo models at both 125M and 1.3B scales, as well as the

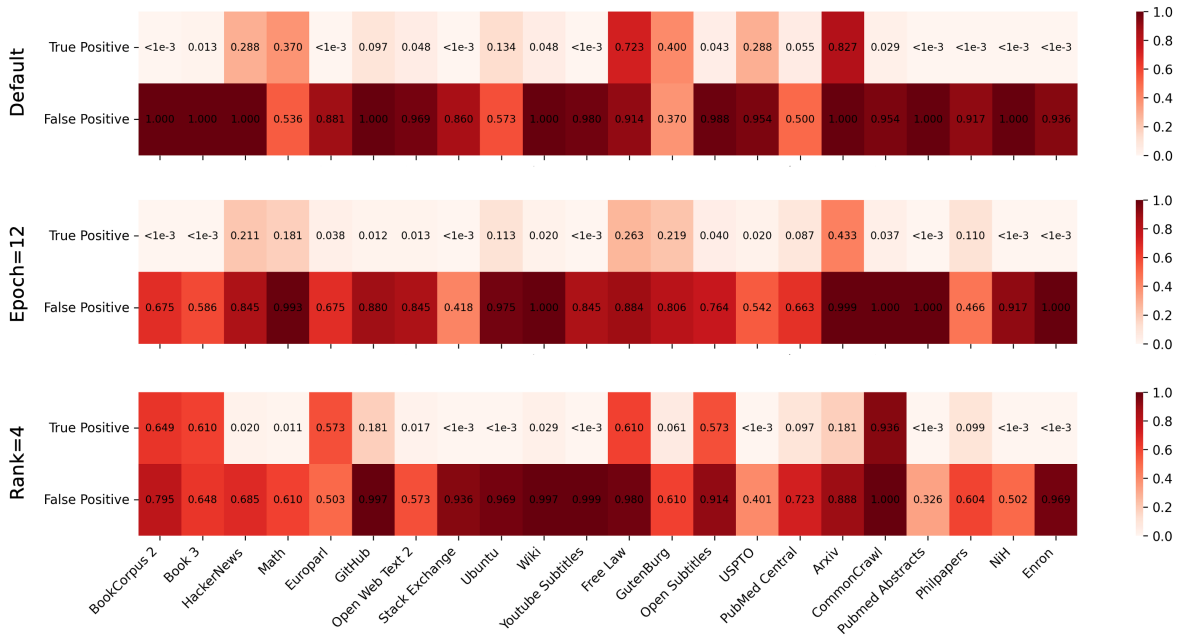

Figure 2: Ablation study. p-values of dataset inference on the Pythia-410M model under different ablation settings. We vary the fine-tuning duration by increasing the number of epochs to 12 and modify the LoRA configuration by reducing the rank to 4, while keeping all other settings unchanged.

20B GPT-NeoX model, where CatShift achieves near-perfect discrimination.

Performance gaps become even more pronounced when considering low false positive regimes. At 5% FPR, CatShift maintains TPRs above 0.95 on Pythia-410M and GPT-Neo 125M, and above 0.82 even on GPT-Neo 1.3B. In contrast, PDD exhibits near-random behavior across all settings, with TPR@5% close to chance level, while LLM DI shows moderate sensitivity but degrades notably on larger models. These results indicate that probability- and divergence-based methods fail to reliably detect membership when restricted to token-level signals.

Overall, CatShift demonstrates both high sensitivity and strong robustness under conservative operating conditions. It consistently detects a large fraction of true members while tightly controlling false positives, and its advantage persists across model families and scales. These findings confirm that catastrophic-forgetting-induced output shifts provide a reliable and scalable signal for dataset inference in logit-inaccessible settings.

Detailed analyses across subsets and failure-case analyses are presented in Appendix C.

4.3 Effectiveness on Commercial Models

In this section, we investigate the effectiveness of CatShift on API-based commercial models, specifically focusing on OpenAI's GPT-3.5. We utilize the fine-tuning interface provided by OpenAI, which is designed to adapt the model to domain-specific tasks by fine-tuning it on custom datasets. For this evaluation, we use the default fine-tuning settings offered by OpenAI, ensuring that the model is trained according to the standard procedure.

However, since the membership status of these models is not publicly disclosed, we refrain from using the PILE dataset for evaluation, as we did in previous sections. Instead, we turn to the BookMIA dataset (Shi et al., 2024), which is specifically designed for evaluating membership status. The BookMIA dataset contains both member and non-member books from OpenAI models, making it an ideal candidate for assessing the membership prediction capabilities of CatShift.

The results show that CatShift achieves a p-value of 6.44×10^{-5} for member books, indicating strong evidence that these books belong to the model's training data. In contrast, the non-member books have a p-value of 0.711, which is significantly higher, demonstrating that CatShift effectively distinguishes between member and non-member data. This clear separation in p-values highlights the robustness and applicability of CatShift when applied to commercial models, such as those offered by OpenAI, further validating its effectiveness in real-world scenarios.

4.4 Ablation Study

We examine the sensitivity of CatShift to key hyperparameters that influence fine-tuning dynamics, focusing on the number of fine-tuning epochs and the LoRA rank.

Impact of Fine-Tuning Epochs. We evaluate the effect of training duration by comparing the default setting of 9 fine-tuning epochs with an extended setting of 12 epochs on Pythia-410M across all 22 PILE subsets. As shown in Figure 2, at a threshold of $p < 0.1$, the epoch=9 configuration correctly identifies 15 member datasets, while epoch=12 identifies 14. Importantly, both settings yield zero false positives, preserving perfect specificity on non-member subsets. These results indicate that modest variations in fine-tuning duration have limited impact on detection performance, and that additional epochs do not necessarily improve sensitivity.

Impact of LoRA Rank. We next study the effect of parameter capacity by varying the LoRA rank, comparing the default rank=8 with a reduced rank=4 under otherwise identical conditions. As illustrated in Figure 2, rank=8 correctly detects 15 member datasets, while rank=4 detects 14 at the same $p < 0.1$ threshold. Both configurations again produce no false positives, indicating that CatShift maintains strong robustness even under more constrained adaptation capacity. This suggests that the method does not rely on aggressive parameter updates and remains effective with lightweight fine-tuning.

Overall, the ablation results indicate that CatShift is not highly sensitive to fine-tuning hyperparameters. Neither increasing the number of epochs nor reducing the LoRA rank leads to meaningful changes in detection accuracy. This suggests that the signal exploited by CatShift stems from relative output shifts caused by catastrophic forgetting, which emerge even under lightweight and conservative fine-tuning. As a result, the method remains effective without careful hyperparameter tuning, making it suitable for practical deployment.

5 Related Work

MIA techniques, first developed for classification models (Shokri et al., 2017), have recently been adapted for LLMs. Likelihood-based metrics such as loss (Yeom et al., 2018) and perplexity (Carlini et al., 2021) are commonly used to distinguish training members from non-members, often yielding high AUC-ROC scores. Min-k%Prob (Shi et al., 2024) extends this idea by averaging the log-likelihood of the lowest-probability $k\%$ tokens in a document. Neighbor-based MIAs refine perplexity scores by comparing the likelihoods across neighboring models M' or documents D' (Duan et al., 2024; Galli et al., 2024). While M'-based approaches require a reference model trained on a disjoint but similar dataset—often unrealistic—D'-based methods are more practical but generally less effective and sensitive to noise parameters.

Recent work (Duan et al., 2024; Maini et al., 2024) highlights biases that inflate reported MIA performance, such as temporal shifts between member and non-member data. Re-evaluation on Pythia (Biderman et al., 2023), which uses unbiased random splits of The Pile (Gao et al., 2021), shows significantly lower AUC-ROC scores. Moreover, simple classifiers can exploit these biases to separate members from non-members (Das et al., 2024; Meeus et al., 2024a), underscoring the need to evaluate MIAs on truly random datasets.

Such evaluation is feasible for open models like Pythia but not for proprietary LLMs that do not disclose training data (e.g., those in copyright disputes). As observed in multiple studies (Duan et al., 2024; Maini et al., 2024; Meeus et al., 2024b), MIA performance varies widely across models and datasets, so results from open LLMs cannot be directly generalized to closed ones.

6 Conclusion

In this paper, we introduced CatShift, a token-only dataset-inference framework that exploits catastrophic forgetting to determine whether a dataset was used in training a large language model. The method fine-tunes the model on part of the suspicious dataset and compares output shifts against those induced by a non-member validation set, enabling reliable detection of proprietary data usage even when log probabilities are inaccessible. Extensive experiments on both open-source and commercial LLMs show that CatShift consistently reduces false positives and false negatives under token-only conditions. These results underscore its practicality for auditing and compliance in real-world scenarios. At the same time, the observed variation across model families highlights opportunities for further study on how architecture and scale influence catastrophic forgetting, pointing to promising directions for advancing token-only dataset inference.

Limitations

Despite the strengths of CatShift, several practical constraints and open questions remain. First, catastrophic forgetting may be weaker or more difficult to detect for certain classes of datasets, especially those that overlap substantially with other data in the model's original training corpus. In such scenarios, it may be challenging to disentangle shifts caused by *true reactivation* of learned knowledge from those arising due to partial redundancy with content already present in the model's parameters. Second, the fine-tuning procedure itself introduces hyperparameters (e.g., learning rate, number of epochs, or choice of layers to update) that can significantly affect the observed output shifts. Determining optimal hyperparameter configurations for membership inference, especially under limited budget constraints, remains an open research question. Finally, while our token-only approach targets modern APIs where log probabilities are withheld, providers may deploy additional strategies (e.g., heavy output truncation or specialized filters) that can further limit the information available to dataset owners, thereby complicating or weakening the method's efficacy.

References

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large-Scale Autoregressive Language Modeling with Mesh-TensorFlow. Software available from Zenodo.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 2633–2650.

Debeshee Das, Jie Zhang, and Florian Tramèr. 2024. Blind baselines beat membership inference attacks for foundation models. *arXiv preprint arXiv:2406.16201*.

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? *arXiv preprint arXiv:2402.07841*.

Adam Dziedzic, Haonan Duan, Muhammad Ahmad Kaleem, Nikita Dhawan, Jonas Guan, Yannis Cattan, Franziska Boenisch, and Nicolas Papernot. 2022. Dataset inference for self-supervised models. *Advances in Neural Information Processing Systems*, 35:12058–12070.

Hao Fang, Yixiang Qiu, Hongyao Yu, Wenbo Yu, Jiawei Kong, Baoli Chong, Bin Chen, Xuan Wang, and Shu-Tao Xia. 2024. Privacy leakage on dnns: A survey of model inversion attacks and defenses. *CoRR*, abs/2402.04013.

Filippo Galli, Luca Melis, and Tommaso Cucinotta. 2024. Noisy neighbors: Efficient membership inference attacks against llms. *arXiv preprint arXiv:2406.16565*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. 2024. Rethinking machine unlearning for large language models. *CoRR*, abs/2402.08787.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *CoRR*, abs/2308.08747.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024. Llm dataset inference: Did you train on my dataset? *Advances in Neural Information Processing Systems*, 37:124069–124092.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. 2021. Dataset inference: Ownership resolution in machine learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. 2024a. Inherent challenges of post-hoc membership inference for large language models. *arXiv preprint arXiv:2406.17975*.

Matthieu Meeus, Igor Shilov, Manuel Faysse, and Yves-Alexandre de Montjoye. 2024b. Copyright traps for large language models. In *41st International Conference on Machine Learning*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Jie Ren, Han Xu, Pengfei He, Yingqian Cui, Shenglai Zeng, Jiankun Zhang, Hongzhi Wen, Jiayuan Ding, Hui Liu, Yi Chang, and Jiliang Tang. 2024. Copyright protection in generative AI: A technical perspective. *CoRR*, abs/2402.02333.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

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 2017, San Jose, CA, USA, May 22-26, 2017*, pages 3–18.

The New York Times. 2023. The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work. https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html.

Johnny Tian-Zheng Wei, Ryan Yixiang Wang, and Robin Jia. 2024. Proving membership in LLM pretraining data via data watermarks. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 13306–13320. Association for Computational Linguistics.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, pages 268–282.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27263–27277.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. 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, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 5263–5274. Association for Computational Linguistics.

A Impact of Fine-Tuning Iterations

In this section, we study how the number of fine-tuning iterations affects CatShift. Figure 3 shows results on the Wikipedia and YouTube Subtitles subsets. As iterations increase, the L2 distance between fine-tuned and original weights grows monotonically, capturing the progressive parameter drift characteristic of catastrophic forgetting. Continued optimization overwrites pre-trained representations, fitting the fine-tuning data more closely. At the same time, p-values from our hypothesis test decrease, indicating that the model's output distribution becomes increasingly distinguishable from that of a non-member dataset. This supports our hypothesis that fine-tuning on member data not only learns new patterns but also reactivates latent knowledge, producing significant shifts in top-1 completions. The inverse relationship between L2 distance and p-value highlights how parameter-level forgetting amplifies output-level membership signals. Our results confirm that the more a model "forgets" during adaptation to member data, the more clearly it "remembers" it, yielding a strong statistical signature for dataset inference.

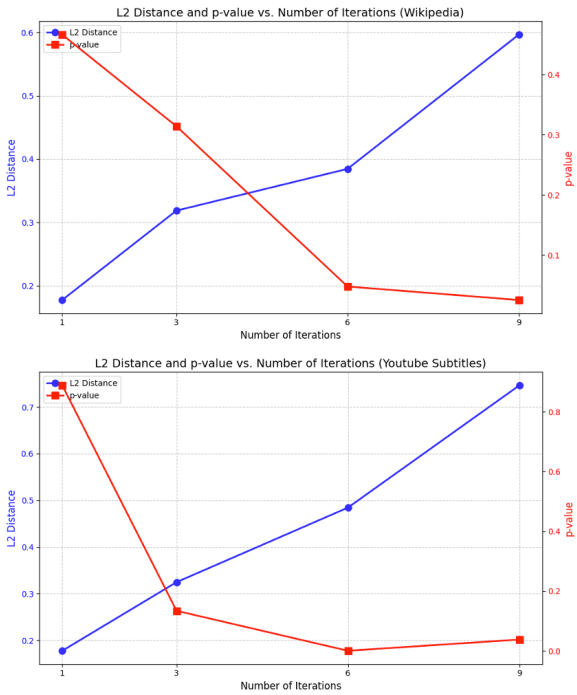

Figure 3: Effect of fine-tuning iterations on CatShift.

B Overall Algorithm

Algorithm 1 Dataset Inference with CatShift

Input: Target dataset $D_s = \{(x_i, y_i)\}_{i=1}^{N}$, target model $f^{(0)}$ with parameters $\theta^{(0)}$
Output: Likelihood of dataset D_s being part of the training data
Split the target dataset D_s into training and test subsets:

$$D_s^{\text{train}}, D_s^{\text{test}} = \text{Split}(D_s)$$

Construct completion function c.
Fine-tune the model $f^{(0)}$ on D_s^{train} to obtain model $f^{(1)}$:

$$\theta^{(1)} = \arg\min_{\theta} \sum_{(x_i, y_i) \in D_s^{\text{train}}} \mathcal{L}(f^{(0)}(x_i), y_i)$$

for each test sample $(x_i, y_i) \in D_s^{\text{test}}$ **do**
  Compute top-1 completion from $f^{(0)}$: $\hat{y}_i^{(0)} = f^{(0)}(c(x_i))$
  Compute top-1 completion from $f^{(1)}$: $\hat{y}_i^{(1)} = f^{(1)}(c(x_i))$
  Compute similarity score $s_i = \text{Sim}(\hat{y}_i^{(0)}, \hat{y}_i^{(1)})$
end for
Compute scores for the known non-member validation set D_v:

$$s_j^v = \text{Sim}(\hat{y}_j^{(0)}, \hat{y}_j^{(1)}) \quad \text{for each } (x_j^v, y_j^v) \in D_v$$

Apply Kolmogorov-Smirnov test: p-value $= \text{KS}(S_s, S_v)$
Decision:
if p-value $< \alpha$ **then**
  Reject the null hypothesis and conclude that D_s is likely part of the model's training data
else
  Fail to reject the null hypothesis and conclude that D_s is not part of the model's training data
end if

Algorithm 1 summarizes the procedure of CatShift. Given a candidate dataset, we first split it into training and test subsets and fine-tune the target model on the training portion using standard completion-style supervision. We then compare the model's behavior before and after fine-tuning by computing top-1 completions for each test prompt under both the original and fine-tuned models. The similarity between these paired completions captures how fine-tuning alters the model's outputs on the candidate data. To determine whether this shift reflects prior exposure or genuine adaptation, we contrast the resulting similarity scores with those obtained from a held-out validation dataset that is guaranteed not to appear in the model's training corpus. Finally, we apply a Kolmogorov–Smirnov test to assess whether the two score distributions differ significantly. A low p-value indicates that fine-tuning induces a structured output shift distinct from that observed on non-member data, providing evidence that the candidate dataset was likely used during training.

C In-Depth Analysis Across Subsets

Beyond aggregate performance, we conduct an in-depth analysis of CatShift at the level of individual PILE subsets to better understand how detection behavior varies across data sources and target models. While Table 1 summarizes overall statistics across multiple architectures, it does not reveal which subsets drive false positives or false negatives. Here, we focus on subset-level p-value distributions to characterize these error patterns in detail.

Figure 4 shows the distribution of p-values for each PILE subset across four representative open-source LLMs. On Pythia-410M, CatShift exhibits extremely conservative behavior. At a threshold of 0.1, no non-member subset is misclassified, indicating strong control of false positives. However, this conservativeness leads to reduced sensitivity: seven member subsets produce p-values above the threshold, corresponding to a false negative rate of 31.8%. This suggests that for smaller models, catastrophic-forgetting-induced shifts may be insufficiently strong or consistent across all training subsets, particularly those with weaker signals.

For the GPT-NEO family, the error profile shifts markedly. The method becomes substantially more sensitive, correctly identifying most member subsets. On GPT-NEO-125M, for example, only three member subsets yield p-values above 10^{-3}, indicating that the majority of training sources are confidently detected. This gain in recall is accompanied by a moderate increase in false positives: at the 0.1 threshold, we observe one false positive on GPT-NEO-125M and GPT-NEO-1.3B, while none occur on GPT-NeoX-20B. These results indicate that more expressive models amplify output shifts on true members, but may also exaggerate spurious similarities for certain non-member subsets.

Failure and Ambiguous-Case Analysis. A closer examination reveals that these errors are not uniformly distributed across subsets. Take Pythia-410M as an example, the highest false-negative rate consistently occurs on the *arXiv* subset. A plausible explanation is that arXiv spans a wide range of scientific disciplines, resulting in substantial heterogeneity in topics, structures, and writing styles.

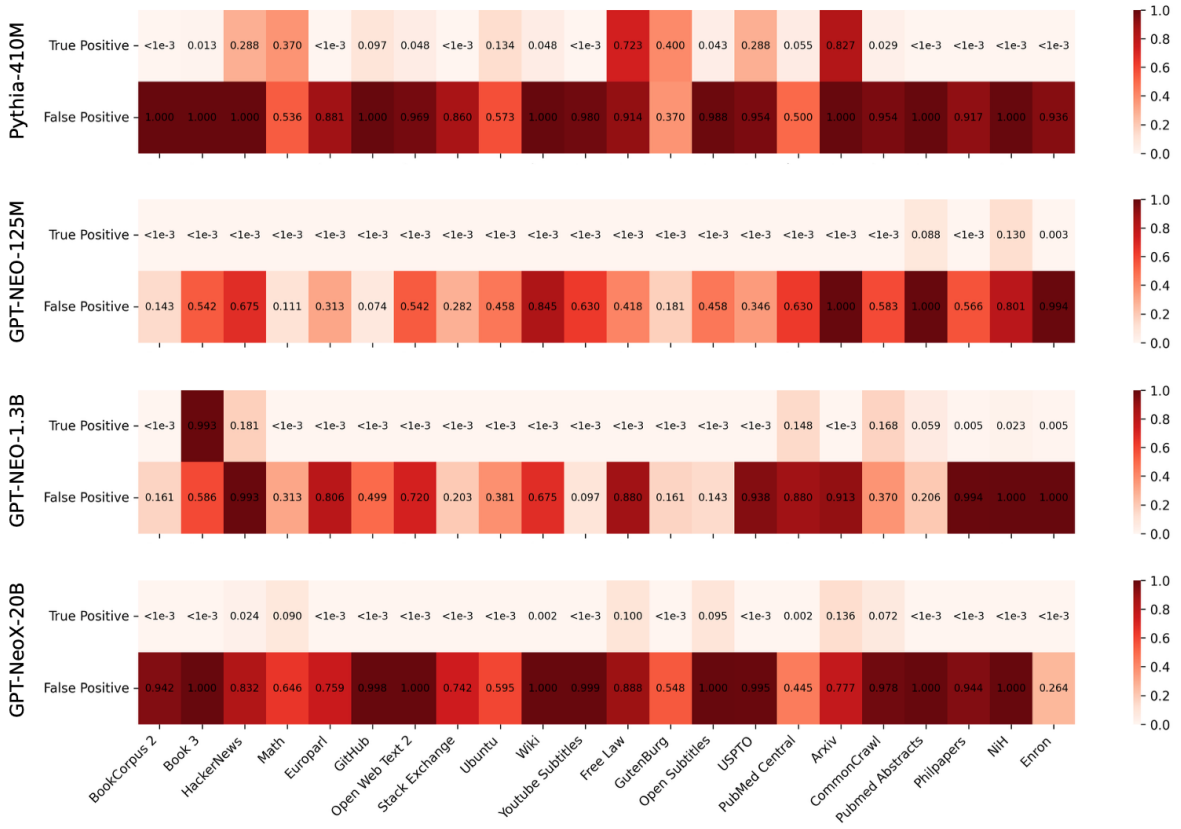

Figure 4: p-values of dataset inference by applying dataset inference to Pythia (Biderman et al., 2023) and GPT-Neo (Black et al., 2021) models with 1000 data points. We can observe that CatShift can correctly distinguish train and validation splits of the PILE with low p-values.

This diversity weakens dataset-level linguistic regularities and reduces the likelihood that the model develops coherent memorization patterns tied to arXiv as a source. As a result, catastrophic forgetting induces weaker and less consistent output shifts, making true membership harder to detect.

In contrast, the highest false-positive rate is observed on the *Gutenberg* subset. Gutenberg consists primarily of long-form literary works with relatively stable narrative structures and stylistic regularities. These properties can strengthen sequence-level memorization tendencies in LLMs and may also induce spurious familiarity effects or hallucinated recall during fine-tuning. Consequently, the detector may over-attribute these stylistic signals to training exposure, leading to inflated false-positive predictions.

D Impact of Similarity Scorer

We study whether CatShift is sensitive to the choice of similarity scorer by comparing the default BERTScore with BartScore (Yuan et al., 2021), which is good at capturing semantic consistency. Experiments are conducted on the GPT-Neo-125M model. The results, shown in Figure 5, indicate that BartScore achieves comparable or slightly improved performance relative to BERTScore. This suggests that CatShift is not tightly coupled to a specific similarity metric and remains effective across different semantic scorers.

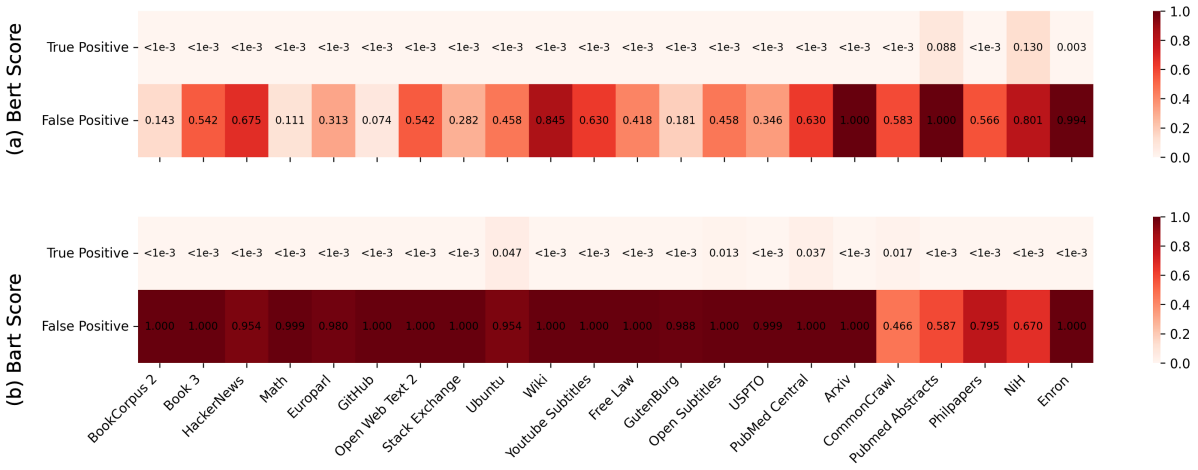

Figure 5: p-values of dataset inference on the GPT-Neo-125M model under different similarity scorers.