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

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have transformed NLP with their ability to interpret, reason, and generate text. Yet their dependence on large, often proprietary datasets raises concerns over unauthorized usage, which can result in copyright infringement and financial harm. Existing dataset-inference techniques typically rely on log probabilities, but many leading LLMs now withhold or perturb these signals—underscoring the need for token-only methods that work without access to internal logits. We propose CatShift, a token-only dataset-inference framework that exploits catastrophic forgetting—the tendency of LLMs to overwrite prior knowledge when learning new data. If a dataset was part of training, fine-tuning on a subset via available fine-tuning APIs triggers a pronounced shift in model outputs, whereas unseen data causes only modest changes. By statistically comparing these output shifts against those from a known non-member validation set, CatShift infers whether the dataset was likely in the model's training corpus. Extensive experiments on both open-source and API-based LLMs demonstrate CatShift's strong performance in logit-inaccessible settings, providing a robust, practical tool for protecting proprietary data.

1 Introduction

Large Language Models (LLMs) demonstrate remarkable capabilities in language understanding, reasoning, and generation, driving advances in tasks such as text classification, sentiment analysis, and question answering. Their success is fueled by web-scale datasets, yet their rapid adoption raises pressing concerns over data rights and licensing.

This growth has intensified debates on unauthorized data use. In particular, copyrighted material may be included in training sets without consent, infringing authors' rights and causing financial harm (Wei et al., 2024; Ren et al., 2024; Liu et al., 2024). For instance, the New York Times recently sued OpenAI and Microsoft over alleged misuse of their content (The New York Times, 2023). Media organizations invest heavily in such datasets and rarely make them public. Even openly released corpora (Raffel et al., 2020; Gao et al., 2021) often restrict usage to research or education. Detecting unauthorized training on proprietary data is therefore 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, commonly via logits or log-probabilities. While effective, such methods fail when models reveal only top-1 predictions, as is already the case for Claude (Anthropic, 2024). This limitation motivates token-only inference, which relies solely on the predicted token. Though broadly applicable and robust to logit perturbation defenses (Fang et al., 2024), naïve token-only baselines (e.g., membership = good completion accuracy) are unreliable, as shown in Section 4.2. More principled strategies are needed.

Our Solution. We leverage catastrophic forgetting (CF) (Luo et al., 2023) to improve membership detection. CF occurs when learning new data overwrites previous knowledge. If a dataset was part of training, fine-tuning on it helps the model "recall" forgotten knowledge, producing a marked pre-/post-tuning output shift. In contrast, unseen data induces smaller changes.

CatShift operationalizes this idea using fine-tuning APIs widely supported by major providers (OpenAI, Hugging Face, Google Vertex AI, Claude). The dataset is split into train/test subsets; the model is fine-tuned on the train portion, and output shifts are measured on the test portion. Greater shifts indicate prior exposure. A held-out validation set unseen by the model provides a non-member baseline.

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 determine whether the distributional shift is significantly larger. A low p-value (e.g., $p < 0.1$) indicates likely membership.

Results. Our method achieves strong performance on both open-source and API-based models. For instance, on Pythia (Biderman et al., 2023), CatShift attains an AUC of 0.979 and F1 of 0.863, surpassing the baseline (AUC 0.707, F1 0.717). 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.

Our results show that token-only inference is feasible without access to logits or privileged signals, enabling practical copyright protection.

Contributions. • We present the first systematic study of dataset inference in token-only settings, addressing real-world scenarios where log probabilities are unavailable.

• We introduce CatShift, a CF-based inference method that consistently outperforms baselines across models and datasets.

• We provide evidence that training members exhibit significantly larger output shifts post-fine-tuning than non-members, explaining why CF-based inference works.

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.

2.3 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 the top-1 predicted tokens rather than logits or probabilities. They can fine-tune the LLM on custom datasets via publicly available APIs and possess a validation set guaranteed not to be in the model's training corpus. This set serves as a baseline for measuring output distribution shifts after fine-tuning on novel data.

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 Methodology of CatShift

3.1 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.

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.

3.2 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.

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 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.

3.3 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.

3.4 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

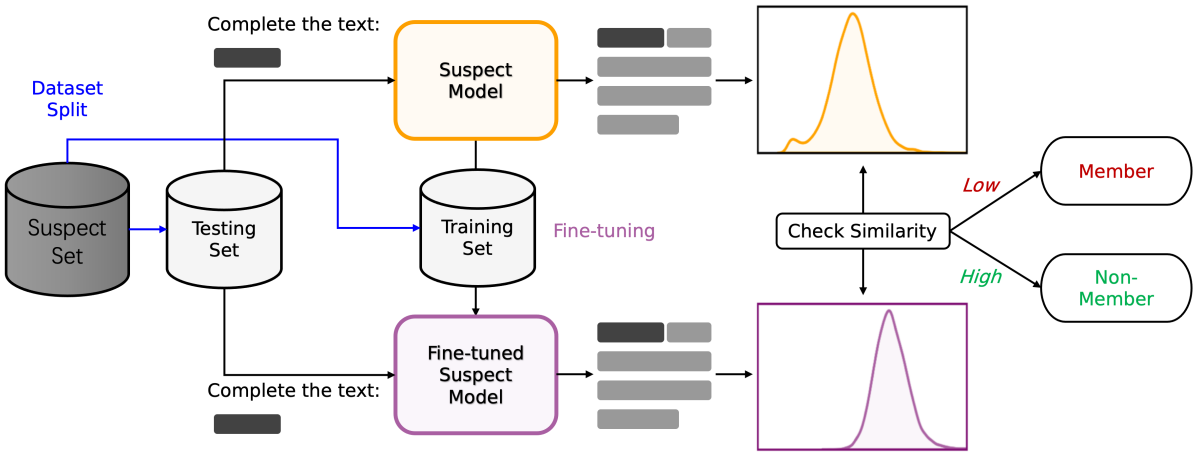

Figure 1: An overview of the CatShift framework for dataset inference on token-only large language models (LLMs). The figure illustrates the three-step process: (1) Dataset Split—where the target dataset is partitioned into training and test subsets; (2) Completion Prompt Construction—where data samples 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.

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.

3.5 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).

Baseline. The baseline does not fine-tune the model but instead measures how well the model completes prompts from the target dataset. For each sample (x_i, y_i), we compute $\hat{y}_i = f(x_i)$ and a similarity score $s_i = \mathrm{Sim}(\hat{y}_i, y_i)$ using BERTScore. The score is the mean similarity

$$\bar{s}_b = \frac{1}{N}\sum_{i=1}^{N} s_i.$$

Higher $\bar{s}_b$ indicates the model likely saw the data during training, whereas lower scores suggest novelty. This provides a direct measure of model familiarity without fine-tuning.

Evaluation Protocol. We run CatShift on each of the 22 PILE subsets, treating training splits as members and validation splits as non-members. We apply the Kolmogorov–Smirnov test to compare the distributions of member and non-member similarity scores:

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

A p-value $p < 0.1$ indicates likely membership. We report ROC AUC and F1 scores (binarizing at $p < 0.1$) to provide a comprehensive view of performance.

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.

Finally, we combine the baseline and CatShift in a dual-test scheme: if the baseline p-value is below 10^{-3}, we accept its decision and skip CatShift; otherwise, we defer to CatShift's result.

4.2 Effectiveness of CatShift

In this section, we evaluate the effectiveness of CatShift on Pythia-410M, comparing it against a simple baseline (Section 4.1) since existing methods

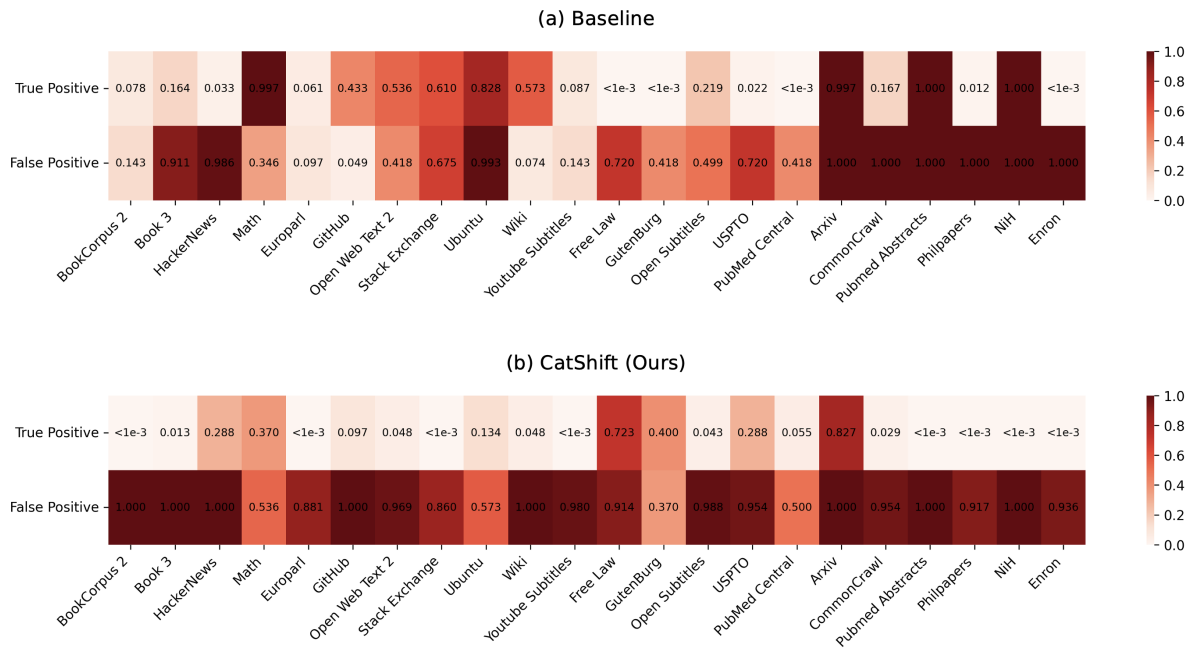

Figure 2: p-values of dataset inference by applying dataset inference to the Pythia-410M model with 1000 data points. We can observe that CatShift can correctly distinguish train and validation splits of the PILE with low p-values (usually below 0.1). Also, when considering false positives for comparing two validation subsets, we observe a p-value higher than 0.1 in all cases, indicating no false positives.

generally assume logit access and are thus inapplicable in our setting.

Figure 2 reports the distribution of p-values across PILE subsets. At a threshold of 0.1, the baseline produces 3 false positives, whereas CatShift produces none. This indicates that our method is more conservative when evaluating non-member subsets and avoids incorrectly attributing validation data to the training corpus. For member subsets, CatShift also shows stronger sensitivity. At the same threshold of 0.1, the baseline misclassifies 12 training subsets as non-members, while CatShift reduces this number to 7. The reduction demonstrates that the output-shift signal exploited by CatShift captures membership information more reliably than simple performance-based heuristics. When adopting stricter conditions, the differences become more pronounced. With a threshold of 1×10^{-3}, CatShift correctly identifies 8 member subsets, compared to only 4 with the baseline. This shows that even under highly conservative significance levels, our method continues to recover signals that the baseline fails to capture.

Aggregate metrics summarize these improvements. The baseline achieves an AUC of 0.707, while CatShift attains an AUC of 0.979, reflecting a substantial gain in ranking quality independent of any fixed threshold. For F1 score, which depends on a decision threshold, we report results at the standard 0.1 level used in the above analyses. Under this threshold, the baseline reaches 0.717, whereas CatShift achieves 0.863. The improvement indicates not only higher precision, driven by the elimination of false positives, but also better recall, due to fewer missed detections of true members.

Overall, CatShift demonstrates both higher specificity and higher sensitivity than the baseline. It eliminates false positives at practical thresholds, detects a greater fraction of true members, and achieves consistently superior performance across conservative and relaxed significance levels. These results establish its effectiveness as a dataset inference method in logit-inaccessible conditions.

4.3 Impact of Target Model

In this experiment, we study how the choice of target model affects the performance of CatShift. We evaluate four representative open-source LLMs of different sizes and architectures: Pythia-410M, GPT-NEO-125M, GPT-NEO-1.3B, and GPT-NeoX-20B.

Figure 3 presents the distribution of p-values for these models. On Pythia-410M, CatShift achieves an extremely low false positive rate, reaching zero

at the 0.1 threshold. However, this strong conservativeness comes at the cost of higher false negatives: seven member subsets are misclassified, corresponding to a false negative rate of 31.8%. This suggests that while CatShift is highly reliable in rejecting non-members for Pythia, its sensitivity to true members is somewhat limited.

On the GPT-NEO models, the trade-off shifts. CatShift demonstrates much stronger sensitivity, with far fewer false negatives. For instance, on GPT-NEO-125M, only three member datasets have p-values above 1×10^{-3}, indicating that most training subsets are correctly detected. This gain in recall, however, is accompanied by higher false positive rates. At the 0.1 threshold, we observe one false positive for GPT-NEO-125M and one for GPT-NEO-1.3B, while GPT-NeoX-20B yields none. The corresponding false positive rates are 4.5% for GPT-NEO-125M, 18.2% for GPT-NEO-1.3B, and 9.1% for GPT-NeoX-20B.

For consistency, we adopt the 0.1 threshold across all models when computing F1 scores. The results highlight both the robustness and adaptability of CatShift across architectures of varying scale. On Pythia-410M, the method achieves an AUC of 0.979 and an F1 score of 0.863. On GPT-NEO-125M, performance improves substantially, with an AUC of 0.994 and an F1 score of 0.957. On GPT-NEO-1.3B, the method attains an AUC of 0.941 and an F1 score of 0.898, slightly lower but still markedly strong. Finally, on GPT-NeoX-20B, CatShift reaches perfect discrimination with an AUC of 1.0 and an F1 score of 0.978.

These results demonstrate that CatShift generalizes well across models of different scales. While smaller models such as Pythia emphasize specificity with very low false positives, larger and more expressive models tend to reduce false negatives and achieve higher overall F1 scores. This pattern suggests that the degree of catastrophic forgetting leveraged by CatShift is influenced by the model's architecture and capacity, with larger models producing more pronounced and consistent output shifts that enhance membership inference.

Performance, however, is not strictly monotonic: GPT-NEO-1.3B shows the highest false positive rate despite being larger than GPT-NEO-125M. This points to the influence of factors beyond model size, such as training data and optimization. Although GPT-NeoX-20B approaches perfect discrimination, results may not generalize to all large models, particularly those with stronger regularization or more diverse corpora. These findings highlight both the robustness and model-dependence of CatShift, motivating evaluation on a wider range of architectures.

4.4 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.

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

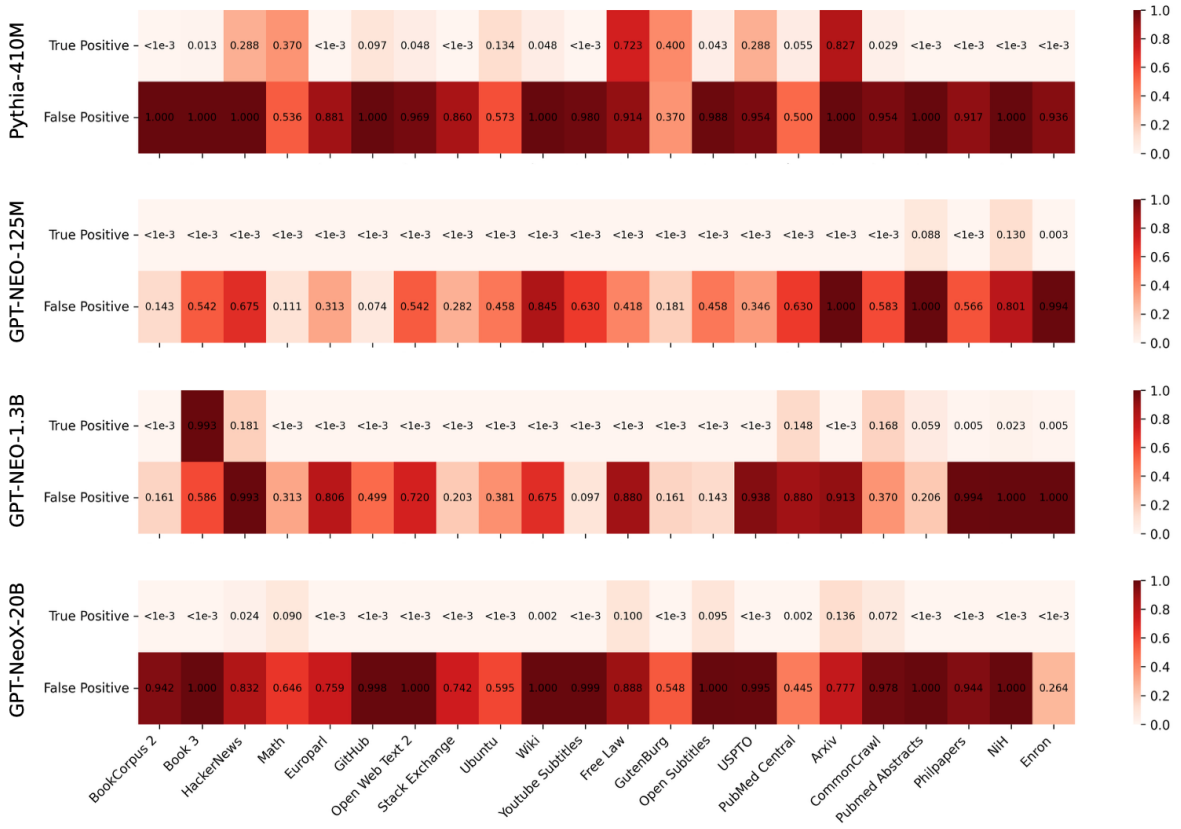

Figure 3: p-values of dataset inference by applying dataset inference to Pythia (Biderman et al., 2023) and GPT-Neo (Black et al., 2021) models with 1000 data points. We can observe that CatShift can correctly distinguish train and validation splits of the PILE with low p-values.

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.

A Impact of Fine-Tuning Iterations

In this section, we study how the number of fine-tuning iterations affects CatShift. Figure 4 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 and detectable statistical signature for dataset inference.

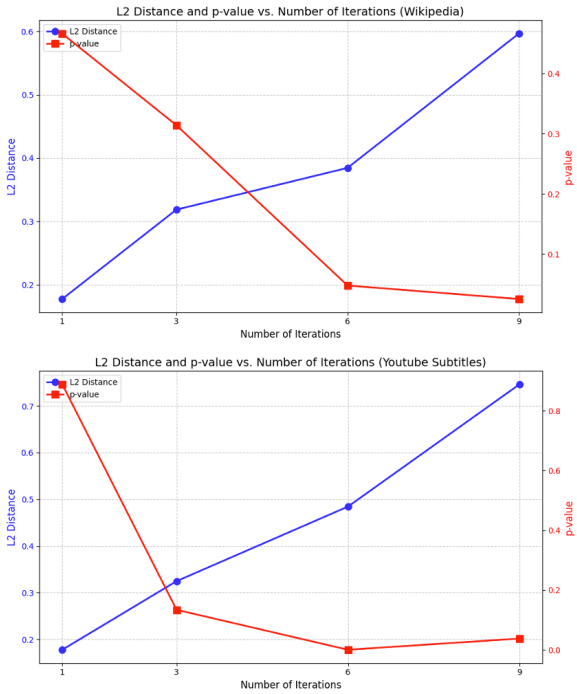

Figure 4: Effect of fine-tuning iterations on CatShift.

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