DATASET INFERENCE VIA CUMULATIVE EVIDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Dataset Inference (DI) aims to determine whether a given dataset was used to train a model. However, existing DI methods for large language models (LLMs) face two significant practical limitations. First, they assume gray-box access to token-level probabilities, whereas most commercial LLMs expose only generated text through black-box APIs. Second, they rely on rigid p-value hypothesis testing, which requires committing in advance to a specific dataset size and significance level. In practice, this makes auditing either too costly or statistically inconclusive, since adaptively adding additional samples invalidates standard p-value guarantees. We address these limitations with a fully black-box and sequential approach to DI. To overcome API restrictions, we introduce a novel method for estimating continuous per-token probabilities directly from discrete, label-only outputs, unlocking **DI in a fully black-box setting**. To address the rigidity of standard fixed-sample testing, we identify that e-values are particularly well suited for this setting. Based on e-values, we create a new sequential testing framework for DI. Crucially, when the initial dataset size is insufficient, our framework seamlessly accommodates additional data collection without violating statistical guarantees, enabling **DI with iterative evidence accumulation**. Together, these advances make DI practical for real-world LLM auditing. We validate our novel DI method on 6 diverse open-source LLMs with known training datasets and demonstrate its efficiency in production settings on 6 LLMs exposed via public APIs. Our cost analysis shows that, for instance, auditing GPT-5 via the OpenAI API using the WikiMIA-24 dataset with our method costs only $1.92.

1 INTRODUCTION

Large Language Models (LLMs) such as GPT (Achiam et al., 2023), Llama (Dubey et al., 2024; Touvron et al., 2023), Gemini (Team et al., 2024), and Claude (Anthropic, 2023) are trained on massive, uncurated datasets. The inclusion of copyrighted content (Cooper et al., 2025) and privacy-sensitive sequences (Carlini et al., 2021) in the training sets of LLMs has raised significant privacy concerns (Rahman & Santacana, 2023; Wu et al., 2023) and triggered numerous copyright infringement lawsuits (Sil, 2023; J. Doe 1 et al. v. GitHub, Inc., Microsoft Corporation, and OpenAI, Inc., 2024; J. Doe 1 et al. v. GitHub., Microsoft, and OpenAI et al., 2024; Maini et al., 2025; Sinai; The New York Times v. Microsoft and OpenAI., 2023). This background sets the stage for the technical challenge of reliably identifying training data within LLMs Maini et al. (2024); Rossi et al. (2026); Zhao et al. (2025).

The first attempts to address the problem used Membership Inference (MI) methods (Shokri et al., 2017; Carlini et al., 2022). MI methods aim to determine whether a specific data point was used to train a suspect model. While effective on smaller classical machine learning models, prior work (Maini et al., 2024; Duan et al., 2024) demonstrated that the immense size of LLM training corpora degrades individual membership inference performance, yielding results that are no better than random guessing. Maini et al. (2024) demonstrated that Dataset Inference (DI) is a much more effective way to solve the problem. Instead of inferring the membership of a single data point, DI aggregates membership signals across multiple examples to infer whether an entire dataset was included in the suspect model's training corpus. However, existing DI methods face two critical limitations that hinder their application to modern commercial LLMs.

First, DI assumes a gray-box threat model, requiring access to model loss values to identify training data (Shi et al., 2023; Zhang et al., 2024). With the release of models like GPT-5 (Singh et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2025), API provides a limited access to the loss values. This restriction renders previous gray-box DI impractical and highlights the urgent need for strictly black-box approaches. Second, current DI methods (Maini et al., 2024; Rossi et al., 2026; Zhao et al., 2025) rely on traditional hypothesis tests to compute a p-value, which measures the likelihood that the observed membership signals occurred purely by chance. The core limitation here stems from the statistical rigidity of p-values which require strict assumptions to remain valid: Notably, the sample size and the significance threshold must be fixed prior to the test. Any attempt to alter them post-hoc or repeatedly test the data invalidates the results (a practice known as p-hacking). This rigidity becomes a significant bottleneck when obtaining membership scores is computationally or financially costly, as an evaluator cannot simply gather more samples to increase statistical power if the initial test yields an insignificant result.

To overcome these limitations and transform DI into a practical tool for identifying training data in modern LLMs, we introduce **D**ataset **I**nference via **C**umulative **E**vidence (**DICE**). As outlined in fig. 1, DICE efficiently adapts prior gray-box membership features (Carlini et al., 2021; Shi et al., 2023; Mattern et al., 2023; Mitchell et al., 2023) into fully black-box counterparts. Crucially, to overcome the statistical rigidity of p-values, DICE abandons traditional testing in favor of a dynamic framework. Specifically, we utilize sequential hypothesis testing by betting (Shekhar & Ramdas, 2023). This approach frames the testing process as a betting game: it continuously places *"bets"* against the null hypothesis, accumulating wealth as evidence grows. This wealth is formalized as an *e-value*: a test statistic whose expected value is at most 1 when the null hypothesis is true. An e-value directly quantifies the strength of evidence against the null; we reject the null when the e-value exceeds $1/\alpha$, which is a valid level-α test.

The fundamental advantage of this sequential framework lies in its theoretical flexibility and practical usability. It allows for the post-hoc addition of data points without invalidating the test results, providing strict theoretical guarantees on false positive control. Because of these properties, sequential testing is perfectly suited for DI. It enables an adaptive, sample-wise analysis where *inference can be terminated at any point without invalidating the statistical guarantees*. In practice, this means an evaluator can begin querying a small subset of the suspect dataset, dynamically add more samples if the evidence remains inconclusive, and immediately stop the process once sufficient confidence is achieved, significantly optimizing the computational and financial costs of the dataset inference.

We validate the correctness of our method on 6 diverse open-source LLMs, including the Pythia family (Biderman et al., 2023), OLMo (Groeneveld et al., 2024), and Qwen (Yang et al., 2024), which were pre-trained on known, diverse training datasets (Gao et al., 2020; Soldaini et al., 2024; Shi, 2023). We also demonstrate the efficiency of DICE in production settings on 6 LLMs exposed via public APIs, including GPT models (Achiam et al., 2023; Singh et al., 2025) and Gemini (Comanici et al., 2025). For example, auditing GPT-5 via the OpenAI Batch API on the WIKIMIA-24 dataset using DICE costs only $1.92. This confirms the practicality of our DICE method for auditing black-box models in real-world scenarios.

Overall, our four main contributions are as follows:

1. We propose DICE, a novel practical DI method for LLMs that operates with only black-box access to the models and provides anytime-valid statistical guarantees.
2. We design novel black-box membership features and translate existing gray-box features to the black-box setup by efficiently estimating per-token probabilities.
3. We equip DICE with a novel statistical testing framework based on e-values, which supports sequential testing, enabling flexible, sample- and compute-efficient inference.
4. We validate our method through comprehensive empirical evaluation on 6 open and 6 closed source LLMs with different sizes, on diverse datasets (including PILE, DOLMA, WIKIMIA-24, and BOOKMIA), and across multiple domains.

2 BACKGROUND AND RELATED WORK

2.1 MEMBERSHIP AND DATASET INFERENCE

Membership Inference (MI). The goal of MI methods is to detect whether a given data point x was used to train a target model (Shokri et al., 2017; Carlini et al., 2022). Multiple MI methods

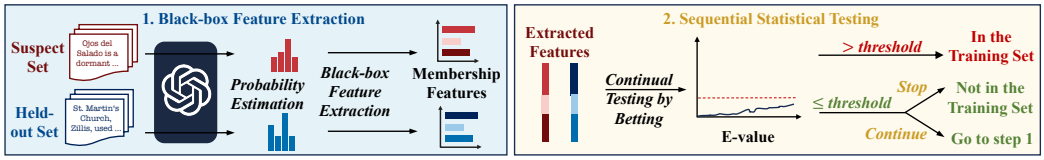


Figure 1: **Overview of DICE: D**ataset **I**nference via **C**umulative **E**vidence. **1. Black-box feature extraction.** We split both suspect and held-out sets into prefixes and suffixes, prompting the LLM exposed via a black-box API with prefixes to generate predicted tokens. Then, we compute semantic similarity between generated tokens and ground-truth suffixes, mapping these to proxy probabilities via the sigmoid mapper to extract membership features. **2. Sequential Statistical Testing.** Using the extracted features, we compute a discrepancy metric and the betting value. Then, we apply sequential statistical testing by betting to accumulate evidence against the null hypothesis. The resulting wealth process is an e-value: it can be monitored continuously, stopped once it crosses the significance threshold, i.e., sufficient evidence is gathered, or safely and adaptively continued with additional data as resources permit.

have been proposed for LLMs. They mainly operate under ***gray-box settings*** and require access to a model's loss or perplexity. Carlini et al. (2021) leverage the observation that the ratio of a model's perplexity on a training sample to the sample's entropy (computed with the zlib library) or to its perplexity under a reference model is typically lower than for unseen texts. Shi et al. (2023) propose MIN-K% PROB based on a simple hypothesis that an unseen text segment is likely to contain K% outlier tokens with low predicted probabilities, while an example seen during training of the model is less likely to contain such outliers. Perturbation-based MI techniques (Mattern et al., 2023) compare a model's perplexity on the original x versus its perturbed version $\tilde{x}$. The perturbations may include synonym substitutions, random deletions, character-level typos, or replacements generated by an external language model (Mitchell et al., 2023). The reasoning is that training set members are more robust to such perturbations than non-members.

More recent ***black-box***, label-only MI methods approximate output probabilities using a surrogate model: DPDLLM fine-tunes a surrogate on the suspect text and uses a text completion task (Zhou et al., 2024), while PETAL estimates the probabilities token by token on a standard GPT-2-XL model as a surrogate (He et al., 2025). However, most existing MI methods for LLMs do not outperform random guessing (Maini et al., 2024; Duan et al., 2024), including even the strong gray-box MIs. Prior studies have attributed their apparent success to distributional shifts between member and non-member data during evaluation (Maini et al., 2024; Duan et al., 2024). This significantly limits the success of MI for identifying training data in LLMs.

Dataset Inference (DI). To overcome the limitation of MI-based methods for training data detection, dataset inference methods for supervised (Maini et al., 2021), self-supervised learning (Dziedzic et al., 2022), and generative models (Maini et al., 2024; Rossi et al., 2026; Zhao et al., 2025) determine whether a given suspect dataset was used to train a model. DI extracts diverse MI-features for individual data points, aggregates them, and applies statistical testing to reliably determine whether the suspect set was used to train the model. However, existing DI methods for generative models face severe limitations in practical applicability. First, they rely on gray-box MI features, whereas state-of-the-art LLMs like GPT, Gemini, or Claude are deployed behind APIs that provide only black-box access (i.e., they only provide generated tokens). Second, prior DI methods rely on statistical testing with p-values which is limited to a fixed size of the suspect set and a predefined significance level. This causes problems since it is not possible to add more data points to the suspect set when the initial test fails without invalidating the statistical guarantees. We overcome these limitations and make DI practical for the state-of-the-art LLMs by extracting black-box features and performing e-value-based statistical testing which provides *anytime-valid* guarantees. We provide more background on the related work in Section G.

2.2 Sequential Statistical Testing by Betting

Sequential statistical testing (Shafer, 2019; Vovk & Wang, 2021) intuitively frames hypothesis testing as a betting game where data arrive sequentially. Consider a hypothesis testing problem with a null hypothesis $H_0 : P \in \mathcal{P}_{\text{null}}$, an alternative $H_1 : P \in \mathcal{P}_{\text{alt}}$, and observations $Z_1, Z_2, \ldots$ lying in some

space $\mathcal{Z}$, drawn i.i.d. according to P. To test H_0, a bettor places repeated bets on the outcomes $\{Z_t : t \geq 1\}$, starting with an initial wealth $W_0 = 1$.

A single round of betting involves two steps. First, the bettor selects a payoff function $\mathcal{S}_t : \mathcal{Z} \to [0, \infty)$. To ensure a fair bet under the null hypothesis, this function is restricted to have an expected value of at most 1 under the null, given all prior observations. Second, the outcome of Z_t is revealed, and the bettor's wealth grows or shrinks by a factor of $\mathcal{S}_t(Z_t)$. After t rounds, the bettor's accumulated wealth is $W_t = W_0 \prod_{i=1}^{t} \mathcal{S}_i(Z_i)$. Under the null, the wealth process $W_t : t \geq 0$ is a nonnegative test supermartingale: conditional on all previous observations, its expected future wealth is no larger than its current wealth. Conversely, if H_1 is true, a well-chosen payoff function ensures the wealth (representing evidence against the null) grows exponentially.

This wealth process naturally yields a sequential test: we reject the null if $W_t \geq 1/\alpha$, where $\alpha \in (0, 1)$ is the desired significance level. Following Shekhar & Ramdas (2023), we adopt the stopping time

$$\tau \ := \ \inf\{t \geq 1 : \ W_t \geq 1/\alpha\},$$

which induces a valid level-α sequential test by Ville's inequality (Ville, 1939). In this framework, the wealth at any round t is an *e-value*, a direct quantification of evidence against the null hypothesis. While p-values are probability-constrained (where smaller values indicate stronger evidence), e-values are expectation-constrained (where larger values indicate stronger evidence). An e-value of 20 can be read as roughly '20-to-1' evidence against the null in the betting interpretation.

The critical advantage of sequential testing for DICE is its *anytime-validity*. Classical hypothesis tests require the sample size and significance threshold to be fixed prior to the experiment; altering data post hoc to achieve significance results in p-hacking. This rigidity is highly problematic in DI, when a predefined set of points might be insufficient to reject the null. Sequential testing resolves this problem by allowing us to continuously gather data and stop the process the moment sufficient evidence is accumulated, all without inflating the Type-I error rate. Moreover, the exponential growth rate of wealth when the alternative hypothesis is true, limits the number of data points required to reach a desired significance threshold, saving computational costs.

3 Our DICE Method

As illustrated in fig. 1, DICE builds upon the canonical DI framework (Maini et al., 2024), operating on two distinct text sets: a suspect set (potential training data) and a held-out set (non-training data). Our pipeline processes data points sequentially across two main stages. In the first stage, we extract a set of Membership Inference (MI) features for each data point in both sets, relying strictly on the generated output text. In the second stage, we perform sequential hypothesis testing to accumulate evidence against the null hypothesis by evaluating the distributional differences between the MI features of the two sets. If sufficient evidence is gathered to reject the null, the process terminates; otherwise, it continues to acquire and evaluate additional data. We detail each component of this pipeline below.

3.1 Black-Box Feature Extraction

For each data point in the suspect and held-out sets, we split the text into a prefix and a suffix. We pass the prefix as input to the black-box target model and collect the predicted tokens. Using the model-predicted tokens and the ground-truth suffix, we extract two complementary sets of black-box MI features: (1) sequence-level and (2) token-level features. By capturing both global and local memorization behaviors, these feature sets act complementarily, where token-level features specifically allow us to adapt traditional gray-box metrics for black-box evaluation.

Sequence-Level Features. We evaluate the target model's ability to complete a given sequence by measuring the semantic similarity between its generated continuation and the true suffix. We extract sequence-level features using two approaches. In the first, *sequence-standard*, the semantic similarity score is used directly as a membership indicator. In the second, *sequence-perturb*, we slightly alter the input text and then measure the semantic similarity between the model's generated continuation and the ground-truth suffix. Member sequences are expected to produce higher baseline similarity scores and to be more sensitive to such perturbations than non-member sequences. More details regarding the utilized perturbations are provided in Section A.3.

Token-Level Features. To extract token-level signals without access to explicit model loss values, we propose an estimation method based on

By substituting these black-box probability estimates for exact probabilities, we successfully adapt a range of established gray-box membership features to the black-box setting. For instance, we adapt MIN-$K\%$ PROB (Shi et al., 2023), which aggregates the log-likelihoods of the $K\%$ least-likely tokens; a high average score strongly suggests the text was included in the pretraining corpus. A comprehensive list of all per-token features utilized in DICE is provided in section A.2.

3.2 Anytime-Valid Sequential Statistical Testing

The sequence-level and token-level features extracted in the previous step form a K-dimensional feature vector for each sample in both the suspect and held-out sets, capturing a rich combination of membership signals. Conventional DI methods (Maini et al., 2024; Dubiński et al., 2025) typically compress such features into a single scalar score. This aggregation is inherently reductive and risks discarding fine-grained information. In contrast, our hypothesis testing framework directly operates on the full K-dimensional feature vectors, preserving these multidimensional signals throughout the testing process.

Our goal is to determine whether the suspect and held-out sets exhibit statistically different feature distributions. To this end, we employ a non-parametric, anytime-valid two-sample test based on betting. At each round t, we must place a bet against the null hypothesis *before* observing the new pair of K-dimensional MI feature vectors $Z_t = (X_t, Y_t)$. To do this, we analyze the historical observations $Z_{1:t-1} = \{(X_i, Y_i)\}_{i=1}^{t-1}$ from the suspect and held-out sets. We compute the discrepancy between the past suspect features $X_{1:t-1}$ and held-out features $Y_{1:t-1}$ using a chosen discrepancy function. Based on this historical discrepancy, we define our betting strategy for round t, allocating a proportion of our current wealth to predict how the next pair will differ. Finally, we observe Z_t. If this new data point exhibits the predicted discrepancy, providing evidence that the underlying distributions are different, our wealth increases. Conversely, if Z_t aligns with the null hypothesis, our wealth decreases. In this context, the wealth is known to be an e-value which quantifies evidence against the null hypothesis.

We assume these pairs are drawn independently from distributions P_X and P_Y, respectively. We then test the null hypothesis $H_0 : P_X = P_Y$ against the alternative $H_1 : P_X \neq P_Y$, i.e., whether the two sets have the same underlying distribution of membership features.

Our anytime-valid sequential test is constructed through a *payoff function*, which determines how evidence accumulates over time. At each round t, the payoff is defined as

$$\mathcal{S}_t(Z_t) = 1 + \lambda_t u_t, \tag{1}$$

where $u_t \in (-1, 1)$ is a bounded discrepancy score measuring how different X_t and Y_t are, and λ_t is the fraction of wealth we "bet" on this discrepancy. Intuitively, u_t captures the strength of evidence in the current observation, while λ_t controls how aggressively we act on that evidence.

To compute the discrepancy u_t, we use Kernel Maximum Mean Discrepancy (Kernel-MMD), a measure that compares entire distributions rather than just simple aggregated statistics. Kernel-MMD works by mapping data into a high-dimensional feature space and measuring how far apart the two distributions are in that space, making it well-suited for complex, multidimensional data. At each step t, we compute a sequential witness function increment $\widehat{h}_t(x_t, y_t)$, the difference of kernel means evaluated on the new observations against the history $\mathcal{F}_{t-1} = \{x_1, \ldots, x_{t-1}, y_1, \ldots, y_{t-1}\}$, which satisfies $\mathbb{E}[\widehat{h}_t \mid \mathcal{F}_{t-1}] = 0$ under H_0. Since $\widehat{h}_t$ is a signed quantity, we set $u_t = \tanh\big(\widehat{h}_t(x_t, y_t)\big) \in (-1, 1)$, ensuring the strict boundedness required by the betting framework.

The bet fraction λ_t is determined by a *betting strategy* that uses only past observations $(Z_1, \ldots, Z_{t-1})$, before seeing the current data point. A good betting strategy aims to grow wealth quickly when the alternative hypothesis is true, while controlling the risk of losing all wealth under the null. To demonstrate the flexibility of our framework, we employ two well-known strategies: Online Newton Step (ONS) (Hazan et al., 2007) and Kelly betting (Kelly, 1956). Both are designed to adaptively adjust bets based on past performance. Further details are provided in section C.2.

In practice, the testing procedure proceeds sequentially. Starting with initial wealth $W_0 = 1$, at each round t we compute the discrepancy u_t, choose a bet λ_t, evaluate the payoff $\mathcal{S}_t(Z_t)$, and update the

cumulative wealth as $W_t = W_{t-1}\mathcal{S}_t(Z_t)$. This wealth can be interpreted as an *e-value*, representing the accumulated evidence against the null hypothesis.

A key advantage of this framework is its *anytime-validity*: the test remains valid no matter when we choose to stop. This allows evaluators to monitor the evidence as data arrive and stop early once sufficient evidence is obtained, without inflating the false positive rate. In addition, the wealth (e-value) provides a direct and interpretable measure of evidence, making the testing process both flexible and statistically rigorous. Further details on these guarantees are provided in section C.3.

4 EXPERIMENTS

We empirically demonstrate that DICE reliably identifies whether a suspect dataset was used to train a given LLM. We evaluate our approach against multiple baselines and across both closed-source LLMs exposed via black-box APIs and diverse open-weight families of LLMs, consistently outperforming the baselines. Finally, we provide a comprehensive ablation study to validate the contribution of each component within our method.

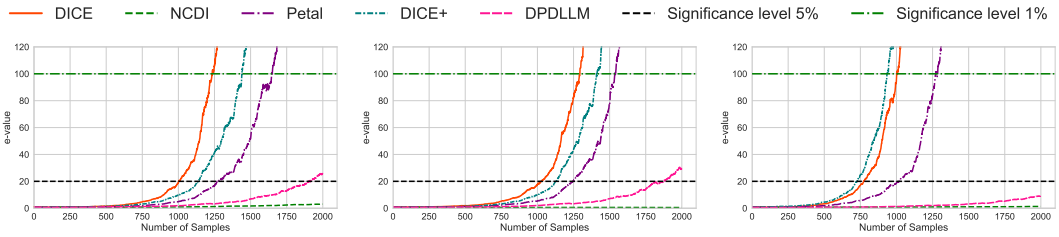
4.1 EXPERIMENTAL SETUP

Models and Datasets. We validate the correctness of DICE on the Pythia family of models (Biderman et al., 2023), specifically PYTHIA-410M, PYTHIA-6.9B, and PYTHIA-12B, evaluated on the PILE dataset (Gao et al., 2020). Furthermore, to demonstrate generalization across diverse model families, we evaluate DICE on OLMO-7B (Groeneveld et al., 2024) using the DOLMA dataset (Soldaini et al., 2024), as well as on the QWEN2-7B and QWEN2-7B-INSTRUCT models (Yang et al., 2024) using the BOOKMIA 2023 dataset (Shi, 2023). More details about the datasets are provided in section A.1. We also evaluate the efficiency of DICE on black-box API models from OpenAI (including GPT-5, GPT-4.1, GPT-4, GPT-4O, and GPT-3.5-TURBO) and from Google (including GEMINI-2.0-FLASH (Team et al., 2023)) using the WIKIMIA-24 dataset (Fu et al., 2025).

Baselines. We evaluate DICE against three baselines: (1) **Naïve Continuation Dataset Inference (NCDI)**, which calculates the similarity between ground-truth sequences and LLM continuations using RoBERTa-Scores; (2) **PETAL** (He et al., 2025), which estimates probabilities via next-token semantic similarities calibrated with a surrogate model. We instantiate PETAL by utilizing its per-sample loss estimate as a MI feature vector; and (3) **DPDLLM** (Zhou et al., 2024), which utilizes GPT-2 as a reference model to fit textual data and acquire sequence probabilities from the generated LLM text. Additionally, we compare against the gray-box counterpart of DICE, the **DICE+** method, which employs the exact same token-level features as DICE but utilizes exact token probabilities directly from the model rather than the black-box estimates. Further details on these baselines are provided in section B. We also tested **CatShift** (Xiong et al., 2025), which measures LLM continuation similarities before and after fine-tuning on the suspect set. However, as CatShift proved largely ineffective, we show its results only in section B.4 and focus our main analysis on NCDI, PETAL, DPDLLM, and DICE+.

Sequential Testing by Betting. To estimate the statistical power of our sequential test, we perform Monte Carlo simulations with $T = 500$ independent trials. In each trial, a data stream of length $N_{\max}$ is generated by randomly sampling without replacement. The sequential test is executed with a fixed rejection threshold of $1/\alpha$. If the accumulated evidence crosses this threshold, we record the stopping time; otherwise, the trial is logged as a non-rejection at $N_{\max}$. For the betting strategy, we impose a stake constraint $\lambda_t \in [-\lambda_{\max}, \lambda_{\max}]$, setting $\lambda_{\max} = 0.5$. Finally, to ensure robustness against data ordering, all experiments are repeated across 25 different random seeds. More information regarding the implementation details of the sequential test is provided in Section C.4.

Performance Metrics. Our primary evaluation relies on *wealth curves* plotted against the number of samples, illustrating the exponential accumulation of evidence against the null hypothesis. To empirically validate our strict Type-I error control, each wealth plot includes a false positive experiment where the suspect set consists entirely of non-members. Additionally, we report the *Average Stopping Time*, which denotes the mean number of observations required to cross the 5% significance threshold across multiple random seeds. From the Monte Carlo simulations of our sequential test, we extract the True Positive Rate (TPR) as the proportion of true-member trials that successfully reject the null and False Positive Rate (FPR) as the proportion of non-member trials that incorrectly reject it.



(a) USPTO on Pythia 12B (b) USPTO on Pythia 6.9B (c) USPTO on Pythia 410M

Figure 2: **Comparison across methods and model sizes.** We compare DICE against the black-box-based PETAL, NCDI and DPDLLM methods as well as the DICE+ baseline (see more datasets in Figure 8).

Crucially, we conduct inference sequentially to overcome the rigidity of traditional methods, which require a fixed, predefined sample set to train a classifier or execute a hypothesis test at a predefined significance threshold. Traditional binary detection approaches typically evaluate performance via a ROC curve, inherently forcing a trade-off where reaching higher True Positive Rates (TPR) demands sacrificing the False Positive Rate (FPR). However, sequential testing with e-values fundamentally circumvents this trade-off by providing strict theoretical guarantees on Type-I error control. Consequently, to align perfectly with the core objective of sequential dataset inference, we evaluate our approach using anytime-valid metrics that are intrinsic to the sequential pipeline, explicitly avoiding reliance on fixed prior sample sizes or static binary classification thresholds.

4.2 Effectiveness of Our Black-Box Features

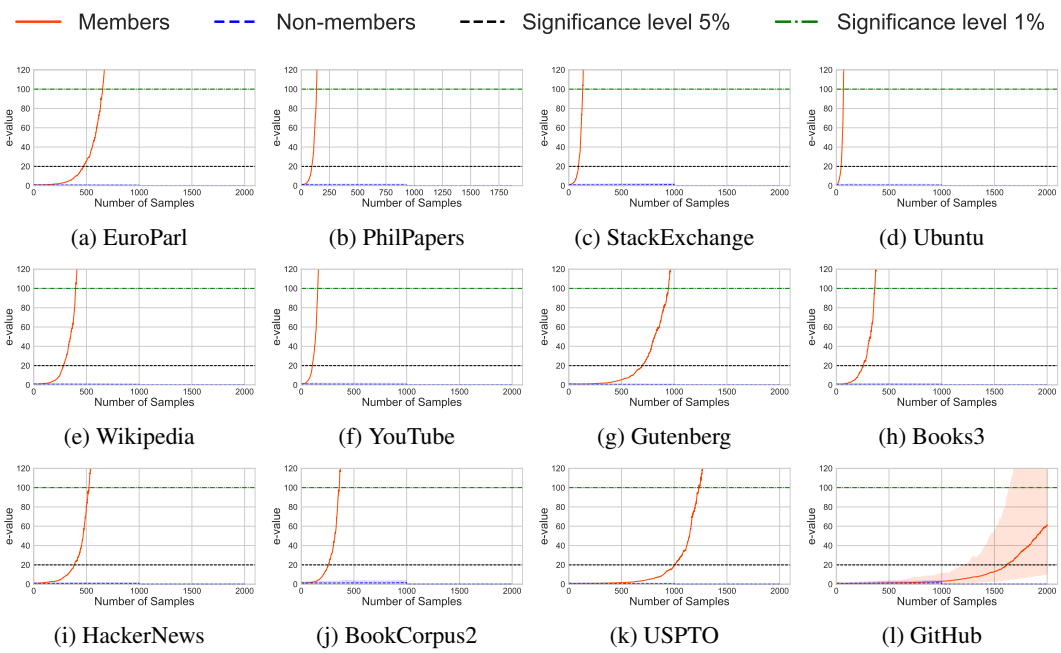
In fig. 2, we evaluate the effectiveness of DICE against competing approaches. The evidence accumulation curves demonstrate that our extracted black-box features and sigmoid-based probability estimates yield robust statistical signals, significantly outperforming both the NCDI and PETAL methods. Furthermore, unlike PETAL, DICE does not rely on a surrogate model to estimate probabilities. This not only makes our approach computationally simpler, but also establishes a stronger signal for dataset identification, a trend that remains consistent across all model scales, from 410M to 12B parameters.

Crucially, fig. 2 also highlights that the performance of DICE remains competitively close to DICE+. This indicates that our proxy features capture sufficient information to drive the sequential test, completely bypassing the need for exact token probabilities. These results suggest that successful dataset inference does not fundamentally depend on privileged access to internal model statistics. Instead, the critical factor is the ability to calibrate membership scores to maximize the distributional separation between member and non-member samples, a task for which our sigmoid-based estimation proves both sufficient and robust. Finally, we provide a comprehensive ablation study in section E.1, where we sequentially remove subsets of the features from the pipeline. The performance drop shows that all of the extracted features are necessary for an optimal performance.

4.3 Effectiveness of Our Testing by Betting

We present the results of our sequential test for the Pythia-12B model on subsets of the Pile dataset, demonstrating the practical effectiveness of testing by betting for Dataset Inference. As depicted in fig. 3, the wealth (e-value) for the suspect set (members) grows exponentially, whereas the wealth for non-members remains near its initial value. This behavior empirically validates our theoretical guarantees: log-optimality ensures rapid, exponential wealth growth for true members, allowing an arbitrator to confidently reject the null hypothesis using significantly fewer observations. Simultaneously, strict Type-I error control rigorously bounds the False Positive Rate (FPR) by preventing any meaningful wealth accumulation when the null is true.

In contrast, prior works relying on traditional p-value testing require predefined, massive sample sizes, often assuming at least 1,000 samples are necessary for verification (Maini et al., 2024). Our framework drastically reduces this requirement. To achieve a 5% significance level, the average stopping times for the Pythia-12B model are merely 93.8 ± 3.9 samples for PhilPapers, $111.8 \pm$



(a) EuroParl (b) PhilPapers (c) StackExchange (d) Ubuntu

(e) Wikipedia (f) YouTube (g) Gutenberg (h) Books3

(i) HackerNews (j) BookCorpus2 (k) USPTO (l) GitHub

Figure 3: **Testing by betting across the Pile subsets.** Accumulated wealth trajectories with 95% confidence intervals (orange shading) for sequential testing by betting on the Pythia-12B model.

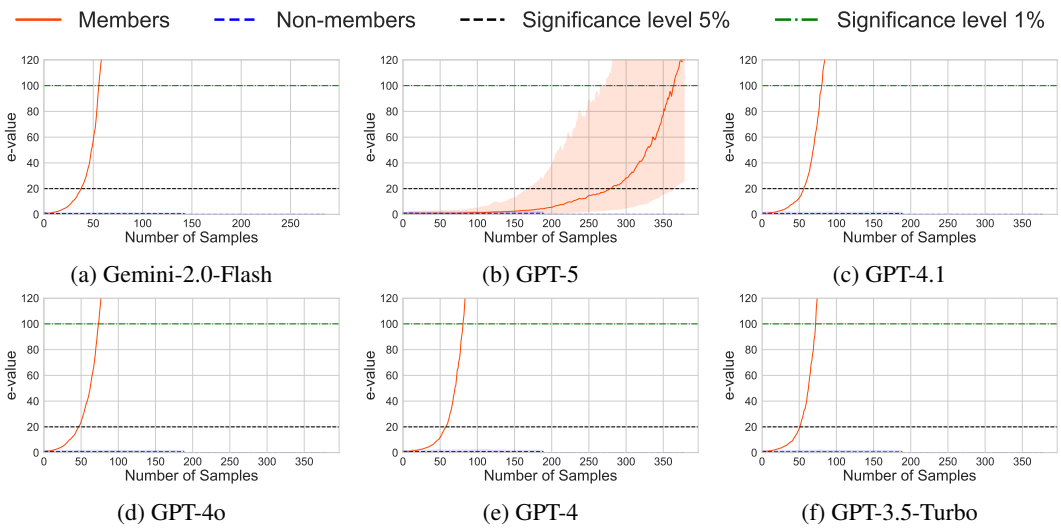
3.4 for StackExchange, 47.0 ± 1.5 for Ubuntu IRC, and 279.7 ± 14.3 for Wikipedia (see table 1 and table 3 for the full results). By minimizing the required data footprint, our method yields substantial savings in the computational resources and API costs associated with feature extraction. Ultimately, the combination of rapid evidence accumulation and anytime-validity makes our framework highly adaptable. It succeeds in data-limited regimes and allows evaluators to safely append new data points post-hoc without the risk of inflating the false positive rate. More results across different methods, model sizes, and model families are provided in section D. We also provide an ablation study for different betting strategies in section E.5 showing that the test performance is robust to the selection of the betting strategy as long as the log-optimality conditions are met.

4.4 Practicality of DICE on Black-Box Models

Having validated DICE on open-source models with known training corpora, we now demonstrate its practicality for auditing state-of-the-art commercial LLMs strictly through black-box APIs. We report the results in Figure 4 for the WikiMIA-24 dataset (Fu et al., 2025) as the suspect set, which uses March 2024 as the cutoff date for distinguishing members from non-members. Due to the proprietary nature of these black-box models, there is an uncertainty involved regarding the exact composition of their training data. Our primary goal is to assess the practicality of DICE in this setting. We selected WikiMIA-24 since its documented knowledge cutoffs make a valid proxy for suspected membership. As shown in Table 1 and Figure 4, our approach consistently succeeds in this restricted setting: it crosses the 1% significance threshold in fewer than 100 samples for most closed models, with the exception of GPT-5, which crosses the threshold with fewer than 300 observations, and maintains uniform FPR control across all observations. On average, our method achieves a TPR of 0.973 with an average stopping time of 89.93 observations for the closed models tested on the WikiMIA-24 dataset. We suspect that the different performance of GPT-5 might stem from its much later release date than the WikiMIA-24 cutoff date. The OpenAI API documentation specifies September 30, 2024, as the model's knowledge cutoff date. Although later snapshots of this model have been released, they are designed to improve performance in specific areas rather than to extend the initial knowledge cutoff. However, this does not undermine our method: as shown in section E.4, our sequential testing pipeline is robust to the contaminated validation set and even in that condition the method successfully finds a significant effect if the alternative hypothesis is true.

Table 1: **Performance of DICE across different models.** We apply DICE to state-of-the-art LLMs. We report true positive rate (TPR), false positive rate (FPR), and average stopping time (mean ± std represented as *Stop*), which are obtained via sequential betting-based hypothesis testing. We apply closed-source Gemini 2.0 Flash (denoted as *Gemini*) and a family of GPT models on the WikiMIA-24 dataset. We also test open-source models (Pythia and OLMo family) on the Wikipedia dataset.

	Closed Source LLMs on WikiMIA-24						*Open Source LLMs on Wikipedia*			
Metric	**Gemini**	**GPT-3.5**	**GPT-4o**	**GPT-4**	**GPT-4.1**	**GPT-5**	**Pythia-410M**	**Pythia-6.9B**	**OLMo-7B**	**Pythia-12B**
TPR	1.000	1.000	0.999	0.999	1.000	0.841	1.000	1.000	0.958	0.999
FPR	0.0006	0.004	0.002	0.0007	0.0008	0.001	0.002	0.006	0.009	0.003
Stop	$37.7_{\pm 2.1}$	$57.1_{\pm 2.5}$	$47.5_{\pm 2.0}$	$66.2_{\pm 3.2}$	$67.3_{\pm 3.2}$	$263.8_{\pm 9.3}$	$221.4_{\pm 9.2}$	$260.0_{\pm 13.6}$	$261.6_{\pm 105.9}$	$279.7_{\pm 14.3}$



(a) Gemini-2.0-Flash (b) GPT-5 (c) GPT-4.1

(d) GPT-4o (e) GPT-4 (f) GPT-3.5-Turbo

Figure 4: **DICE for black-box LLM APIs.** We present the wealth trajectory for WikiMIA-24 dataset with Gemini and GPT models.

4.5 Black-Box API Cost Analysis

To demonstrate the practical efficiency of our method, we provide a comprehensive cost analysis detailing the required API calls and the associated financial expense to reach statistical significance. This analysis confirms that our sequential testing framework effectively leverages early stopping to maintain exceptionally low operational costs. For example, on the WikiMIA-24 dataset, our sequential test achieves statistical significance after processing only 264 samples for the GPT-5 model. In total, reaching the sequential stopping point necessitates 35,376 API calls, translating to approximately 2.32 million input tokens and 92,600 output tokens. Based on standard GPT-5 pricing ($1.25 per 1M input tokens and $10.00 per 1M output tokens), successfully detecting dataset membership costs a mere $3.83. If executed via the OpenAI Batch API, this cost drops to just $1.92. We present the detailed analysis in Section E.8.

5 Discussion and Conclusions

DICE is a novel black-box and adaptive DI framework for detecting whether a given dataset was used to train an LLM. Our key innovation is the integration of black-box membership features with sequential and anytime-valid statistical testing, enabling us to perform dataset inference for any LLM using a flexible betting protocol that delivers valid results after any number of evaluated samples. This approach significantly reduces the computational burden of black-box dataset inference by allowing early stopping as soon as sufficient evidence is obtained, or continuation for increased statistical evidence as resources permit. Our empirical evaluation, performed on diverse models trained on various datasets, confirms the effectiveness and scalability of our approach. Overall, DICE provides a practical black-box solution for real-world model auditing, supporting data owners' ability to control how their datasets are used to train LLMs.

AI USE STATEMENT

LLMs were used during the preparation of this manuscript for editing and refining the writing. They are not involved in any scientific contributions, including the design of the method, the theoretical framework, the experimental setup, and the interpretation of results.

ETHICS STATEMENT

We have read and agree to adhere to the ICLR Code of Ethics. Our work addresses the problem of identifying whether a dataset was used to train an LLM, which has direct implications for data governance and intellectual property protection. DICE is designed as an auditing tool that empowers data owners to verify unauthorized use of their data. All experiments involving commercial LLMs were conducted through their publicly available APIs in accordance with each provider's terms of service. The datasets used in this work (Pile, Dolma, WikiMIA-24, and BookMIA) are publicly available benchmarks. No human subjects were involved in this research.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of all components necessary to reproduce our results. The experimental setup, including model specifications, dataset splits, and hyperparameter choices, is described in section 4.1. The complete list of membership inference features is given in section A.2, with perturbation details in section A.3 and token probability estimation methods in section A.4. Our sequential testing framework, including the betting strategies and their implementation, is fully specified in section 3.2 with additional details in section C.2 and section C.4. All datasets used are publicly available on Hugging Face, with exact repository links provided in section A.1.

REFERENCES

Sarah Silverman and authors sue OpenAI and Meta over copyright infringement. 2023. URL https://www.nytimes.com/2023/07/10/arts/sarah-silverman-lawsuit-openai-meta.html.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anthropic. Introducing Claude. https://www.anthropic.com/index/introducing-claude, 2023. Accessed: 2024-06-01.

Akshay Balsubramani and Aaditya Ramdas. Sequential nonparametric testing with the law of the iterated logarithm, 2016. URL https://arxiv.org/abs/1506.03486.

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

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

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914, 2022. doi: 10.1109/SP46214.2022.9833649.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

A Feder Cooper, Aaron Gokaslan, Amy B Cyphert, Christopher De Sa, Mark A Lemley, Daniel E Ho, and Percy Liang. Extracting memorized pieces of (copyrighted) books from open-weight language models. *arXiv preprint arXiv:2505.12546*, 2025.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jan Dubiński, Antoni Kowalczuk, Franziska Boenisch, and Adam Dziedzic. CDI: Copyrighted Data Identification in Diffusion Models. In *The IEEE CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2025.

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

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Mia-tuner: adapting large language models as pre-training text detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27295–27303, 2025.

Jean-loup Gailly and Mark Adler. Zlib compression library. 2004.

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

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

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

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.

Yu He, Boheng Li, Liu Liu, Zhongjie Ba, Wei Dong, Yiming Li, Zhan Qin, Kui Ren, and Chun Chen. Towards label-only membership inference attack against pre-trained large language models. In *USENIX Security*, 2025.

J. Doe 1 et al. v. GitHub, Inc., Microsoft Corporation, and OpenAI, Inc. Complaint, No. 4:22-cv-06823 (N.D. Cal. filed Jan. 25, 2024). https://storage.courtlistener.com/recap/gov.uscourts.cand.403220/gov.uscourts.cand.403220.200.0.pdf, 2024. U.S. District Court for the Northern District of California, Oakland Division.

J. Doe 1 et al. v. GitHub., Microsoft, and OpenAI et al. Order Granting in Part and Denying in Part Motions to Dismiss, No. 4:22-cv-06823-JST (N.D. Cal. July, 2024). https://insideaipolicy.com/sites/insideaipolicy.com/files/documents/2024/jul/ai07082024_4.pdf, 2024. U.S. District Court for the Northern District of California.

John L Kelly. A new interpretation of information rate. *the bell system technical journal*, 35(4): 917–926, 1956.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. 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, 2021.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. LLM dataset inference: Did you train on my dataset? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Pratyush Maini, Joseph C. Gratz, and A Feder Cooper. Data privacy, memorization, and legal implications in generative ai: A practical guide, 2025. URL https://neurips.cc/virtual/2025/loc/san-diego/109588.

Tudor Manole and Aaditya Ramdas. Martingale methods for sequential estimation of convex functionals and divergences. *IEEE Transactions on Information Theory*, 69(7):4641–4658, July 2023. ISSN 1557-9654. doi: 10.1109/tit.2023.3250099. URL http://dx.doi.org/10.1109/TIT.2023.3250099.

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

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. DetectGPT: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Noorjahan Rahman and Eduardo Santacana. Beyond fair use: Legal risk evaluation for training LLMs on copyrighted text. 2023. URL https://genlaw.org/CameraReady/57.pdf.

Aaditya Ramdas and Ruodu Wang. *Hypothesis Testing with E-values*. Cambridge University Press, 2025.

Aaditya Ramdas, Johannes Ruf, Martin Larsson, and Wouter M Koolen. Admissible anytime-valid sequential inference must rely on nonnegative martingales. *arXiv preprint arXiv:2009.03167*, 2020.

Lorenzo Rossi, Bartłomiej Marek, Franziska Boenisch, and Adam Dziedzic. Natural identifiers for privacy and data audits in large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=doaAUf9Pi7.

Glenn Shafer. The language of betting as a strategy for statistical and scientific communication. *arXiv preprint arXiv:1903.06991*, 2019.

Glenn Shafer. Testing by betting: A strategy for statistical and scientific communication. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 184(2):407–431, 2021.

Shubhanshu Shekhar and Aaditya Ramdas. Nonparametric two-sample testing by betting. *IEEE Transactions on Information Theory*, 70(2):1178–1203, 2023.

Weijia Shi. Bookmia dataset. *huggingface*, 2023. URL https://huggingface.co/datasets/swj0419/BookMIA.

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

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

Tany Sinai. Dilemma: Recent lawsuits and implications. URL https://www.darrow.ai/resources/ai-copyright-law.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15725–15788, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

The New York Times v. Microsoft and OpenAI. Complaint, No. 1:23-cv-11195 (S.D.N.Y. filed Dec. 27, 2023). https://hh-law.com/wp-content/uploads/2024/07/New-York-Times-complaint.pdf, 2023. U.S. District Court for the Southern District of New York.

Yao Tong, Jiayuan Ye, Sajjad Zarifzadeh, and Reza Shokri. How much of my dataset did you use? quantitative data usage inference in machine learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=EUSkm2sVJ6.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models, 2023.

Jean Ville. *Étude critique de la notion de collectif*. Collection des monographies des probabilités. Paris, 1939. URL https://www.numdam.org/item/THESE_1939__218__1_0.pdf. Doctoral thesis, Université de Paris.

Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *The Annals of Statistics*, 49(3):1736–1754, 2021.

Xiaodong Wu, Ran Duan, and Jianbing Ni. Unveiling security, privacy, and ethical concerns of ChatGPT. *Journal of Information and Intelligence*, 2023.

Chen Xiong, Zihao Wang, Rui Zhu, Tsung-Yi Ho, Pin-Yu Chen, Jingwei Xiong, Haixu Tang, and Lucila Ohno-Machado. Hey, that's my data! label-only dataset inference in large language models, 2025. URL https://arxiv.org/abs/2506.06057.

Jianxin Yang et al. Qwen2 technical report. 2024. URL https://arxiv.org/abs/2407.10671. arXiv preprint.

Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=sT7UJh5CTc.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.

Bihe Zhao, Pratyush Maini, Franziska Boenisch, and Adam Dziedzic. Unlocking post-hoc dataset inference with synthetic data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=a5Kgv47d2e.

Baohang Zhou, Zezhong Wang, Lingzhi Wang, Hongru Wang, Ying Zhang, Kehui Song, Xuhui Sui, and Kam-Fai Wong. DPDLLM: A black-box framework for detecting pre-training data from large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 644–653, 2024.

A Implementation Details

A.1 Dataset

We utilize the Pile dataset sourced from the Hugging Face repository `pratyushmaini/llm_dataset_inference`, matching the exact splits used by Maini et al. (2024). We evaluate on 18 subsets of the Pile dataset. Each subset contains approximately 2,000 member samples (used to train the Pythia suite) and 2,000 non-member samples (unseen during pretraining), with each sample comprising roughly 512 tokens. For true-positive experiments, the member data serves as the suspect set, and the non-member data serves as the held-out set. To simulate false-positive scenarios, we randomly split the non-member data in half, assigning one half as the suspect set and the other as the held-out set. We specifically exclude the Enron and NIH subsets of Pile because they contain fewer than 1,000 samples and lack sufficient sequence length (fewer than 64 tokens). We apply an identical experimental setup to the Dolma dataset (`allenai/dolma`), which we use to evaluate the OLMo models. For our real-world black-box API experiments, we utilize two additional datasets. The WikiMIA-24 dataset is obtained from `wjfu99/WikiMIA-24`. The BookMIA dataset is sourced from `swj0419/BookMIA`, where we dynamically adjust the member and non-member labels according to the specific knowledge cutoff dates of the target models. Finally, across the experiments with open source models, we fix the ground-truth suffix length to 64 tokens when computing membership features based on BERT-Score and for the experiments with the black-box models we use 96 tokens as the prefix.

A.2 Per-token Black-Box Membership Features

We utilize the following per-token black-box features:

1. **Perplexity.** A fundamental baseline feature that computes the perplexity (or average loss) of the sequence on the target model.
2. **MIN-$K\%$ PROB (Shi et al., 2023).** Computes the average log-likelihood of the $K\%$ least-likely tokens in the text. A high average score strongly suggests that the text was included in the pretraining corpus. In our black-box setting, we empirically find that setting $K = 60$ (i.e., 60%) yields optimal performance.
3. **MAX-$K\%$ PROB.** Computes the average log-likelihood of the $K\%$ most-likely tokens in the text. Unlike MIN-$K\%$ PROB, which focuses on the model's uncertainty at its least confident tokens, this feature captures the upper end of the token probability distribution. Member sequences tend to exhibit a higher concentration of tokens assigned very high probability by the model, reflecting stronger memorization of frequently occurring patterns in the training data.
4. **Perturbation-Based (Mattern et al., 2023; Mitchell et al., 2023).** Quantifies the change in perplexity between the original text and a perturbed variant (e.g., generated via synonym substitution, random deletion, or paraphrasing by an auxiliary language model). Member sequences typically exhibit higher sensitivity to these perturbations.

5. **Reference-Model-Based.** Evaluates membership by comparing the suspect model's perplexity on the given sequence against the perplexity of an independent baseline reference model.
6. **Zlib Ratio (Carlini et al., 2021).** Calculates the ratio between the target model's perplexity and the entropy (in bits) of the text after `zlib` compression (Gailly & Adler, 2004). This metric relies on the intuition that if a model was trained on a specific data point, it will encode that information much more efficiently than a generic, out-of-the-box compression algorithm.

A.3 Text Perturbations

For each prefix, we generate six perturbed variants using the NL-Augmenter library. The perturbations include:

1. **Synonym Substitution.** Words are replaced with their synonyms at a probability of 0.25, preserving the overall semantic meaning of the text.
2. **Butter Fingers.** Simulates human typing errors by randomly replacing characters with their keyboard neighbors at a probability of 0.1. For example, the character 'g' might remain unchanged or be swapped with 't', 'b', 'f', 'h', 'e', 'd', 'c', 'y', 'j', or 'n'.
3. **Random Deletion.** Words are randomly removed from the text with a probability of 0.25. This introduces incomplete information, testing whether the model can still infer the missing context.
4. **Change Character Case.** Letters are randomly toggled between uppercase and lowercase at a probability of 0.1 (e.g., 'which' $\rightarrow$ 'WhIch').
5. **Whitespace Perturbation.** Spaces are randomly removed (probability 0.1) or artificially inserted (probability 0.05). This disrupts token boundaries and formatting without altering the underlying characters (e.g., 'because the problem' $\rightarrow$ 'be ca use thep roblem').
6. **Underscore Trick.** Spaces are randomly replaced with underscores at a probability of 0.25 (e.g., 'Hello world' $\rightarrow$ 'Hello_world').

A.4 Methods for Token Probability Estimation

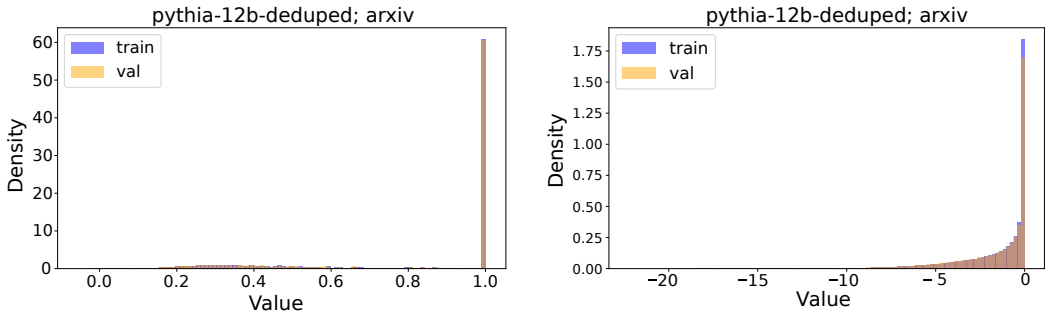
To adapt gray-box membership features to the black-box regime, we emulate exact token probabilities using semantic similarity. Given a prefix $x_1, \ldots, x_{t-1}$, we use greedy decoding to generate the predicted next token $\hat{x}_t$. We then extract embeddings for both the ground-truth token x_t and the predicted token $\hat{x}_t$ using the `all-MiniLM-L6-v2` model, computing their cosine similarity $\mathrm{sim}(x_t, \hat{x}_t)$.

As illustrated in fig. 5, raw similarity distributions differ significantly from actual log-probabilities. Most notably, similarity heavily spikes at 1.0 when the target model predicts the exact correct token (which occurs in roughly 50% of cases across our datasets). To address this distributional shift and reliably estimate proxy token probabilities $\tilde{p}$, we evaluated three distinct transformations:

1. **Raw Similarities.** Because higher semantic similarity (approaching 1) naturally corresponds to lower loss (approaching 0), the simplest approach is a direct linear mapping:
$$-\tilde{p}(\hat{x}_t \mid x_1, \ldots, x_{t-1}) = 1 - \mathrm{sim}(x_t, \hat{x}_t) \tag{2}$$
2. **Reference Model.** Following the original PETAL method, this approach requires an auxiliary reference model (e.g., GPT-2 XL) to extract oracle token probabilities. A linear regression is then fitted against the per-token similarities to learn a slope (α) and intercept (β), estimating the probability as:
$$\tilde{p}(\hat{x}_t \mid x_1, \ldots, x_{t-1}) = \alpha \cdot \log(\mathrm{sim}(x_t, \hat{x}_t)) + \beta \tag{3}$$
3. **Sigmoid (Ours).** The sigmoid function $\sigma(\cdot)$ naturally maps any real-valued input to the bounded interval $(0, 1)$, making it theoretically ideal for modeling probabilities directly:
$$\tilde{p}(\hat{x}_t \mid x_1, \ldots, x_{t-1}) = \sigma\big(\mathrm{sim}(x_t, \hat{x}_t)\big) \tag{4}$$

As demonstrated in fig. 6, our sigmoid-based approach significantly outperforms the alternatives. By mapping similarity values to a strictly bounded range, it inherently dampens the influence of extreme

perplexity outliers and prevents excessive compression during feature normalization, yielding the most robust signal for downstream dataset inference. We note that during our preliminary investigations, we utilized a linear model to aggregate the extracted membership features into a single scalar score, subsequently applying a standard two-sample t-test to calculate traditional p-values. While the findings from this initial approach are provided for contextual reference, our finalized methodology entirely circumvents the scoring model. Consequently, all primary results presented throughout this paper are derived directly using our framework for sequential testing by betting.



(a) Similarity distribution

(b) Log-probability distribution

Figure 5: **Comparison of similarity and log-probability distributions.** We use the Pythia-12B model on ArXiv subset of Pile for this experiment.

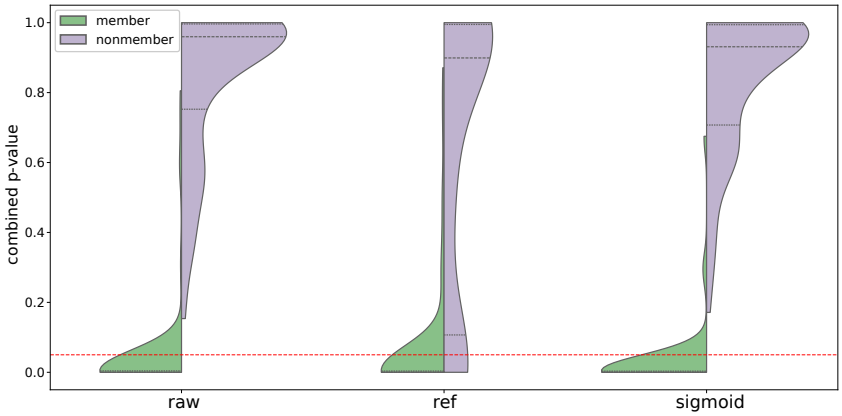


Figure 6: **Distribution of p-values across all Pile subsets on Pythia-12B.** For each subset, we run the scoring model 40 times with 1000 samples per run. The resulting p-values are grouped in batches of four and combined within each group. The violin plots display the distribution of these combined p-values aggregated from all subsets for three different methods: raw similarities, reference model (based on PETAL), and our approach based on the sigmoid function. The better method has lower p-values for members (shown in green) and higher p-values for non-members (shown in purple).

A.5 Normalization and Trimming of Extreme Values

After extracting the per-token and per-sequence membership features from the suspect and held-out sets, and estimating the proxy probabilities via our sigmoid-based calibration (as detailed in section A.4), we perform normalization and trimming before passing the features to the sequential testing process. Since the test proceeds sequentially, these preprocessing steps are also computed sequentially using only observations available up to the current round.

To ensure the features are on a comparable scale, we apply Z-score normalization. For a given feature, let $\{t_j\}_{j=1}^{n}$ denote the sequence of metrics extracted from the suspect set up to round n, and let $\{v_j\}_{j=1}^{n}$ denote the corresponding sequence from the held-out set. At round n, we compute the

empirical mean and standard deviation of the suspect metrics observed *strictly prior* to the current round:

$$\mu_{n-1} = \frac{1}{n-1}\sum_{j=1}^{n-1} t_j, \tag{5}$$

$$\sigma_{n-1} = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n-1}\big(t_j - \mu_{n-1}\big)^2}. \tag{6}$$

We then normalize the current suspect and held-out metrics using these statistics:

$$\tilde{t}_n = \frac{t_n - \mu_{n-1}}{\sigma_{n-1}}, \tag{7}$$

$$\tilde{v}_n = \frac{v_n - \mu_{n-1}}{\sigma_{n-1}}. \tag{8}$$

Normalization is bypassed for early rounds in which $\sigma_{n-1} = 0$ or fewer than $n_{\min}$ observations have been collected, avoiding division by zero. Here, $\tilde{t}_n$ and $\tilde{v}_n$ represent the normalized feature values for the suspect and held-out samples at round n, respectively. The same suspect-set statistics are used to normalize both values, ensuring that the two streams remain on a common scale throughout the sequential test.

Finally, to mitigate the adverse effects of extreme outliers on the downstream sequential test, we apply trimming at each round. Specifically, using the normalized values observed up to round n, we identify the bottom 2.5% and top 2.5% of the distribution and replace these extreme values with the empirical mean of the normalized suspect values. In early rounds, when too few observations are available to estimate these quantiles reliably, trimming is bypassed. In our ablation study in section E.3, we empirically show that this sequential normalization and trimming procedure improves stability and accelerates evidence accumulation against the null hypothesis.

A.6 Black-Box API Evaluation

We apply DICE to black-box large language model (LLM) APIs using the WikiMIA-24 dataset and detail the evaluation protocol below. WikiMIA-24 adopts March 2024 as its knowledge cutoff date for distinguishing member and non-member samples. To determine the corresponding knowledge cutoff dates of the evaluated LLMs, we rely on publicly available documentation, primarily this repository, which collects dates from official technical reports, API providers, GitHub issues, and other public resources.

Notably, the WikiMIA-24 cutoff date precedes the reported knowledge cutoff for GPT-5. Consequently, a fraction of the validation set may contain samples that are members of GPT-5's training data. However, as shown in Section E.4, DICE is robust to polluted validation sets: even when a subset of validation samples are members, the method remains effective, although additional observations may be required to reach a fixed significance threshold. This effect likely explains the reduced performance observed for GPT-5 in our experiments.

An additional consideration for GPT-5 is its strong instruction tuning. When providing a prefix-only input via OpenAI's Responses API, the model may generate meta-level or instructional outputs rather than a direct text continuation. To mitigate this behavior, we prepend the following prompt to each prefix before querying the model:

> "You are a text-completion engine. Continue the given text seamlessly. Do not add commentary, titles, explanations, quotes, or extra formatting. Output ONLY the continuation."

This prompting strategy is applied exclusively to GPT-5. Other models, including GPT-4, GPT-4o, and GPT-4.1, are queried using the Chat Completions API, while GPT-3.5-Turbo is accessed through the legacy Chat Completions API. Despite these interface differences and the strong instruction-following behavior of newer models, we successfully recover membership signals using DICE by framing queries as text continuation tasks, demonstrating the applicability of DICE in realistic black-box API settings. We provide an example of a prefix, ground-truth suffix and generations by each black-box LLM in Table 2.

Table 2: **Example generations for a fixed prefix.** We show the ground-truth suffix and the continuations generated by each black-box LLM when prompted with the same prefix (text-completion setting).

Field / Model	Text
Prefix	`Christina Victoria Grimmie (March 12, 1994 - June 10, 2016) was an American singer and YouTuber. In 2009, she began posting covers of popular songs onto YouTube. After releasing her debut EP, Find Me (2011), her YouTube channel reached one million subscribers. After she reached two million subscribers, she released her debut studio album, With Love (2013). Grimmie then became a contestant on season six of The Voice, finishing in third place. Adam Levine, her coach on the show, announced in the finale that, regardless of the results, he would sign her to his label 222 Records. Lil Wayne also offered`
Ground-truth suffix	` to sign her to his label, Young Money Entertainment. She was signed with Island Records for a short time before being dropped. Grimmie released her second EP,`
Gemini-2.0-Flash	`her a contract with his label, Young Money Entertainment. She ultimately chose to sign with Island Records and released her second EP, Side A (2016`
GPT-3.5-Turbo	`to sign her to his label Young Money Entertainment.\n\nGrimmie continued to release music independently, including her second EP, Side A (2016), which was`
GPT-4	`to sign her to his label, Young Money Entertainment. However, Grimmie decided to sign with Island Records. \n\nGrimmie's music was influenced by contemporary`
GPT-4.1	`to sign her to his label, Young Money Entertainment, following her impressive performances on The Voice. Despite these offers, Christina continued to pursue her music career independently,`
GPT-4o	`to sign her to his label, Young Money Entertainment. Christina Grimmie's performances on The Voice garnered her significant attention and praise, further boosting her music career.\n\nFollowing`
GPT-5	`to sign her to his Young Money Entertainment label; however, neither deal ultimately materialized. Following The Voice, Grimmie released her second EP, Side A`

B Implementation Details for Baseline Methods

To evaluate the effectiveness of our approach, we compare it against four black-box baselines: (1) Naïve Continuation Dataset Inference (NCDI), (2) PETAL, (3) DPDLLM, and (4) CatShift. Furthermore, to quantify the performance gained by having access to exact token probabilities, we include a fifth, gray-box baseline: DICE+. Below, we detail the implementation of each method.

B.1 Naïve Continuation Dataset Inference (NCDI)

NCDI detects memorization by evaluating the degree to which an LLM accurately reproduces the ground-truth continuation of a given sequence. Specifically, we compare the model-generated suffix to the original suffix using the RoBERTa-Score—a variant of BERT-Score (Zhang et al., 2020) that utilizes the RoBERTa-large model (Liu et al., 2019). Higher similarity values are intuitively interpreted as stronger evidence that the sequence was memorized during training.

In practice, each test example is split into a 448-token *prefix* and a 64-token *suffix*. We prompt the target model with the prefix to generate exactly 64 subsequent tokens. We then calculate the RoBERTa-Score between this generated prediction and the ground-truth suffix. Finally, we input the resulting membership features into the sequential testing pipeline.

B.2 PETAL

The PETAL method (He et al., 2025) estimates next-token probabilities in a black-box setting by leveraging a surrogate model M_s. The implementation follows a four-step pipeline:

1. **Surrogate Prediction and Similarity Calculation.** For each token position t, given a prefix $p = x_1, x_2, \ldots, x_{t-1}$ and the ground-truth next token x_t, we query the surrogate model M_s (restricted to generate a single token) to obtain the predicted token $\hat{x}_t = M_s(p)$ and its exact log-probability

$\log P_s(x_t \mid p)$. We then extract embeddings for both x_t and $\hat{x}_t$ using the `all-MiniLM-L6-v2` model. Let these embeddings be $\mathbf{e}(x_t)$ and $\mathbf{e}(\hat{x}_t)$. Their cosine similarity is computed as:

$$\text{sim}(x_t, \hat{x}_t) = \frac{\mathbf{e}(x_t) \cdot \mathbf{e}(\hat{x}_t)}{\|\mathbf{e}(x_t)\| \|\mathbf{e}(\hat{x}_t)\|}$$

2. **Linear Calibration.** We fit a linear calibration function by regressing the surrogate log-probabilities onto the logarithmic similarities. The parameters α and β are learned via least squares:
$$\log P_s(x_t \mid p) = \alpha \log\big(\text{sim}(x_t, \hat{x}_t)\big) + \beta$$
3. **Target Model Estimation.** To apply this to the black-box target model M_t, we prompt M_t with the identical prefix p to generate $\hat{x}'_t = M_t(p)$. We compute the similarity $\text{sim}(x_t, \hat{x}'_t)$ and apply the learned calibration parameters to estimate the target log-probability:
$$\log \tilde{P}_t(x_t \mid p) = \alpha \log\big(\text{sim}(x_t, \hat{x}'_t)\big) + \beta$$
4. **Aggregation and Hypothesis Testing.** For each sequence in both the suspect set D_{sus} and the held-out set D_{hold}, we aggregate the estimated log-probabilities over T tokens:
$$\bar{\ell} = \frac{1}{T} \sum_{t=1}^{T} \log \tilde{P}_t(x_t \mid x_{<t})$$
Finally, we perform the non-parametric two-sample testing by betting to do a hypothesis test on the distributions of $\bar{\ell}$ over D_{sus} and D_{hold}. This yields an e-value for each subset, the distributions of which are visualized in fig. 8 with comparison to other black-box methods and different model sizes.

B.3 DPDLLM

Given a target model f_θ trained on seen data D_s, the goal of DPDLLM (Zhou et al., 2024) is to learn a detector $h(x, f_\theta) \to \{0, 1\}$ for samples $x \in D_s \cup D_u$, where D_u denotes unseen data. For a given text sequence $T = (x_1, \ldots, x_N)$, DPDLLM first fine-tunes a smaller reference model g_ϕ (which, in our implementation, is GPT-2 117M) to memorize the target texts using the standard language modeling objective:

$$\max_{\phi} \sum_{i=1}^{N} \log p(x_i \mid x_1, x_2, \ldots, x_{i-1}).$$

Next, a prefix of the target text (which in our evaluation consists of 16 tokens),

$$T' = (x_1, x_2, \ldots, x_{\lfloor N/2 \rfloor}),$$

is submitted to the black-box LLM to produce a completed text:

$$\hat{T} = f_\theta(T') = (x_1, \ldots, x_{\lfloor N/2 \rfloor}, \hat{x}_{\lfloor N/2 \rfloor + 1}, \ldots, \hat{x}_N).$$

The underlying intuition is that if T appeared in the LLM's pretraining data, the generated continuation $\hat{T}$ should closely resemble the original text. DPDLLM then feeds $\hat{T}$ into the fine-tuned reference model to extract token-level conditional probabilities as features:

$$O = g_\phi(\hat{T}) = \{o_i \mid i = 1, \ldots, N\}, \qquad o_i = p(\hat{x}_i \mid \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{i-1}).$$

Assuming these probability features follow class-conditional Gaussian distributions, the authors train a Gaussian Naive Bayes classifier with parameters

$$\tau = \{(\mu_k, \sigma_k) \mid k \in \{0, 1\}\},$$

estimated from labeled training examples. Membership is then predicted by maximizing the posterior probability:

$$\tilde{y} = \arg\max_c \log p(y = c \mid O) = \arg\max_c \log \left(p(y = c) \prod_{i=1}^{N} p(o_i \mid y = c) \right).$$

In the original methodology, the reference model is fit on a training set D_{train} to collect LLM-generated continuations, extract probability sequence features, and train the classifier. During evaluation, a separate reference model is fit on the test set D_{test} prior to feature extraction.

To ensure compatibility with our sequential hypothesis testing framework and to facilitate a fair baseline comparison, our implementation bypasses this classifier training step. Instead, we directly extract the probability metrics for both the suspect and held-out sets and feed them into our sequential testing pipeline. It is important to emphasize that DPDLLM fundamentally assumes access to both member and non-member training data. In contrast, DICE operates under a significantly less restrictive assumption, requiring access only to a non-member held-out set.

B.4 CatShift

CatShift (Xiong et al., 2025) leverages the phenomenon of catastrophic forgetting, where models overwrite previously learned representations during fine-tuning but can quickly "reactivate" them when re-exposed to familiar data. Specifically, CatShift partitions a suspect dataset into a training subset and a testing subset. It then fine-tunes a target LLM on the training portion using a standard text completion task. Model outputs on the testing subset are collected both before and after fine-tuning, and the extent of their change is quantified using metrics such as BERT-Score. The distribution of these similarity scores is then compared against those from a known non-member held-out set using statistical tests (e.g., the Kolmogorov-Smirnov test). If the suspect dataset was included in the pretraining corpus, fine-tuning induces a more pronounced output shift due to the reactivation of forgotten knowledge, whereas completely novel data induces only modest changes.

In our implementation, we use a suspect dataset comprising 1,000 samples, evenly split into a fine-tuning set D_{train} and an evaluation set D_{test}. The base model M_{base} is fine-tuned on D_{train} for 10 epochs to produce the adapted model M_{ft}. We also utilize a held-out set D_{hold} containing 500 samples, which are drawn from the same underlying distribution but were strictly excluded from the original training of M_{base}.

For each test sequence, we construct a prefix p such that the ground-truth suffix s consists of exactly 64 tokens. Both M_{base} and M_{ft} are prompted with p (restricted to generate a maximum of 64 tokens), and their respective outputs, y_{base} and y_{ft}, are recorded. We then compute the semantic similarity $\text{BERT}(y_{\text{base}}, y_{\text{ft}})$ for each sample in both D_{test} and D_{hold}. To determine whether fine-tuning induces a statistically significant behavioral shift, we perform a one-sided two-sample t-test comparing the similarity distributions of D_{test} and D_{hold}, yielding a p-value at each epoch.

As shown in fig. 7, CatShift performs exceptionally poorly in our black-box setting. It identifies only a single true positive after the 1st and 3rd epochs, and peaks at merely three true positives by the 10th epoch—drastically underperforming even the simple NCDI Baseline, which successfully detects 8 true positives under the same conditions. Because CatShift exhibited poor performance during our preliminary investigations, we excluded it from the sequential testing pipeline. Consequently, we report its evaluation results here utilizing traditional p-values rather than e-values.

B.5 DICE+

DICE+ is a baseline that incorporates the same set of per-token membership features as DICE. The key distinction is that DICE+ directly uses the model's true per-token log-probabilities, rather than relying on calibrated estimates as in the black-box setting. This baseline is designed to quantify the performance gains attributable to access to gray-box per-token probability information. All other implementation details, including feature extraction and downstream inference procedures, are shared between DICE+ and DICE.

C Nonparametric Two-Sample Testing by Betting

An *e-value* is a nonnegative statistic whose expectation under the null hypothesis is at most one. Intuitively, it quantifies evidence against the null: large e-values suggest the null is implausible, while small values are consistent with it. Following Shafer (2021); Shekhar & Ramdas (2023), hypothesis testing can be viewed as a *betting game* against the null. At each round t, the arbitrator observes a pair of data points $Z_t = (X_t, Y_t)$ and selects a payoff function S_t with null-conditional expectation

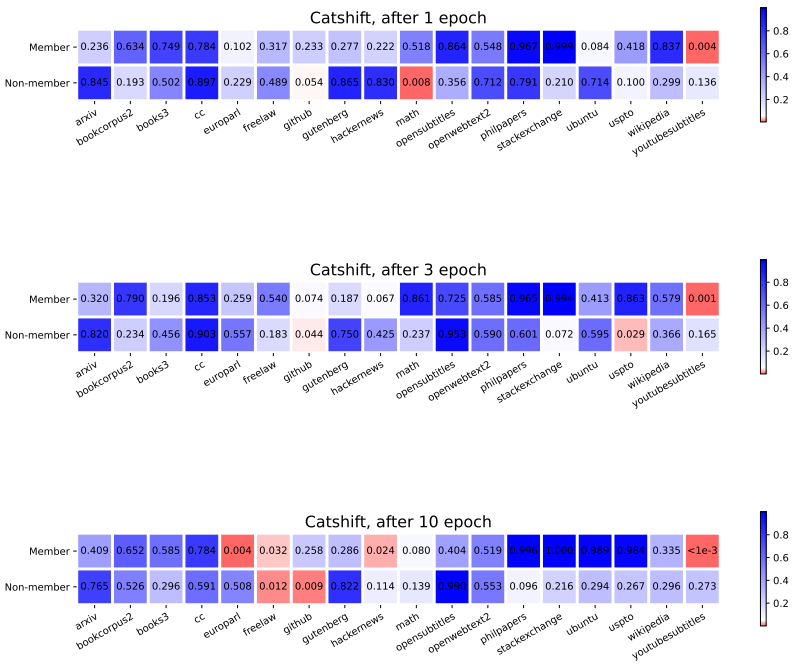


Figure 7: **Epoch-wise p-values for the CatShift method.** We show the values after fine-tuning Pythia-410M (after the first, third, and tenth epochs). Similar trends are observed across intermediate epochs.

at most one. The bettor's wealth is updated multiplicatively as

$$W_t \;=\; W_{t-1}\, S_t(Z_t), \quad W_0 = 1. \tag{9}$$

The resulting sequence $\{W_t\}_{t\geq 0}$ is a non-negative supermartingale under H_0, known as an *e-process* (Ramdas et al., 2020; Ramdas & Wang, 2025). By Ville's inequality (Ville, 1939), exceeding a threshold (e.g., $W_t \geq 1/\alpha$) guarantees a valid level-α test at any stopping time. Thus, the current wealth W_t itself serves as an e-value, providing a calibrated measure of evidence against the null hypothesis. Such an approach introduces numerous advantages over traditional p-value based testing, in particular intuitive properties allowing for simple interpretation of the results.

Sequential adaptivity, theoretical Type I error control, and post hoc interpretability are the main reasons we incorporated Nonparametric Two-Sample Testing by Betting into our pipeline. This method removes the need for a fixed prior sample set for hypothesis testing and allows additional data points to be incorporated sequentially after the test has started. More importantly, unlike classical fixed-sample detection metrics that are typically summarized by a TPR–FPR trade-off, the betting-based method provides an anytime-valid guarantee: under the null, the test wealth forms an e-process whose expectation is at most one at every time. As a result, Type I error can be controlled even under optional continuation, while improved power is typically achieved through more evidence or longer observation rather than through an explicit increase in FPR.

C.1 Discrepancy Metric

The payoff function, which drives the multiplicative wealth updates at each step, relies on its discrepancy component to quantify the distributional distance between the observed data streams. In our experiments, we evaluate Kernel Maximum Mean Discrepancy (Kernel-MMD) as our discrepancy metric. This selection is mathematically motivated: it not only strictly preserves the theoretical guarantees of non-parametric testing by betting, but its inherent kernel structure also facilitates the direct computation of discrepancies across the K-dimensional MI feature vectors. This capability

entirely circumvents the need for an intermediate score aggregation step, thereby preventing additional complexity in the pipeline architecture.

Kernel-MMD. Kernel Maximum Mean Discrepancy (MMD) is the foundation of a widely used batch two-sample test (Gretton et al., 2012). Sequential nonparametric counterparts have also been developed (Balsubramani & Ramdas, 2016; Manole & Ramdas, 2023). In this work, we adopt the sequential Kernel-MMD construction proposed by Shekhar & Ramdas (2023, Sec. 4) as part of our prediction strategy.

To formalize this test, we first consider a general distance measure over a class of test functions $\mathcal{G}$:

$$d_{\mathcal{G}}(P_X, P_Y) = \sup_{g \in \mathcal{G}} \left| \mathbb{E}_{P_X}[g(X)] - \mathbb{E}_{P_Y}[g(Y)] \right|.$$

The function $g^* \in \mathcal{G}$ that achieves this supremum is called the *witness function*, as it provides the maximum contrast between the distributions P_X and P_Y. However, calculating the exact witness function requires explicit knowledge of the underlying data-generating distributions. Because these distributions are unknown in practice, we must design a distance measure that empirically approximates this witness function using a suitably constrained function class.

By restricting $\mathcal{G}$ to the unit ball of a Reproducing Kernel Hilbert Space (RKHS), we obtain the Kernel-MMD. Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel, and let $\mathcal{H}_K$ be its corresponding RKHS. The RKHS is the completion of finite linear combinations of $\{K(x, \cdot) : x \in \mathcal{X}\}$ equipped with an inner product satisfying the reproducing property:

$$g(x) = \langle g, K(x, \cdot) \rangle_{\mathcal{H}_K}, \qquad \langle K(x, \cdot), K(y, \cdot) \rangle_{\mathcal{H}_K} = K(x, y).$$

Specializing the test function class $\mathcal{G}$ to the RKHS unit ball ($\|g\|_{\mathcal{H}_K} \leq 1$) yields the formal definition of MMD:

$$\mathrm{MMD}(P_X, P_Y) = \sup_{\|g\|_{\mathcal{H}_K} \leq 1} \left| \mathbb{E}_{P_X}[g(X)] - \mathbb{E}_{P_Y}[g(Y)] \right| = \|\mu_{P_X} - \mu_{P_Y}\|_{\mathcal{H}_K},$$

where $\mu_P := \mathbb{E}[K(X, \cdot)] \in \mathcal{H}_K$ denotes the kernel mean embedding of a given distribution P.

In the sequential setting, rather than computing the global MMD distance (which is non-negative and unsuitable for betting), we evaluate an incremental *witness function* at each step t. Given the history $\mathcal{F}_{t-1} = \{x_1, \ldots, x_{t-1}, y_1, \ldots, y_{t-1}\}$, the increment is

$$\widehat{h}_t(x_t, y_t) = \frac{1}{t-1} \sum_{s=1}^{t-1} \big[K(x_t, x_s) - K(x_t, y_s) - K(y_t, x_s) + K(y_t, y_s) \big]. \tag{10}$$

This quantity is a signed, antisymmetric estimate of the witness function evaluated on the new pair (x_t, y_t). Under H_0, symmetry of the kernel gives $\mathbb{E}[\widehat{h}_t \mid \mathcal{F}_{t-1}] = 0$. The discrepancy score is then $u_t = \tanh\big(\widehat{h}_t(x_t, y_t)\big) \in (-1, 1)$, which preserves the sign and satisfies the boundedness required by the betting framework.

C.2 Betting Strategy

The second core component of the payoff function is the betting strategy. It determines the stake, the proportion of accumulated wealth to wager on the outcome of the next observation, based on the historical discrepancy between the data streams observed thus far. We evaluate two distinct betting strategies: Online Newton Step (ONS) and Kelly Betting.

Crucially, to maintain valid Type-I error control under the null hypothesis, any stake λ_t wagered at round t must be strictly $\mathcal{F}_{t-1}$-measurable. In measure-theoretic terms, $\mathcal{F}_{t-1}$ represents the natural filtration (i.e., the complete history of all observations and computed discrepancies) up to step $t-1$. Therefore, the stake for any given round can only be derived using data observed prior to that round.

Online Newton Step (ONS). Under ONS, after observing the data pair Z_t at round t, we compute a bounded discrepancy score $u_t \in (-1, 1)$ (e.g., derived from the Kernel-MMD witness function increment). We then update a one-dimensional curvature accumulator a_t and calculate the subsequent stake λ_{t+1} for round $t+1$:

$$z_t = \frac{u_t}{1 + \lambda_t u_t}, \qquad a_t = a_{t-1} + z_t^2, \tag{11}$$

$$\lambda_{t+1} = \Pi_{[-\lambda_{\max}, \lambda_{\max}]}\left(\lambda_t + \frac{2}{2 - \log 3}\,\frac{z_t}{a_t}\right), \tag{12}$$

with the initializations $\lambda_1 = 0$ and $a_0 = 1$. Here, $\Pi_{[-\lambda_{\max}, \lambda_{\max}]}$ denotes the projection onto the permissible, bounded stake interval.

Kelly Betting. Originally introduced by Kelly (1956), this strategy allocates bets proportionally to the empirical strength of the evidence, aiming to maximize the expected logarithmic wealth. This principled allocation theoretically achieves the highest possible long-term exponential growth rate of wealth. In practice, we employ a quadratic approximation of the Kelly criterion. To strictly preserve $\mathcal{F}_{t-1}$-measurability, the stake λ_{t+1} for round $t+1$ is computed using only the discrepancy scores u_j observed up to round t:

$$\lambda_{t+1} = \Pi_{[-\lambda_{\max}, \lambda_{\max}]}\left(\frac{\sum_{j=1}^{t} u_j}{\sum_{j=1}^{t} u_j^2 + \varepsilon}\right), \tag{13}$$

where $\varepsilon > 0$ is a small constant added to the denominator to ensure numerical stability during the early rounds of testing.

C.3 Anytime-Validity and Post-Hoc Significance

Let $\mathcal{P}_0 = \{(P_X, P_Y) : P_X = P_Y\}$ denote the composite null hypothesis. For any (possibly data-dependent) stopping time τ adapted to the natural data filtration, and for any predictable betting strategy used to form the wealth process W_t, the e-process property combined with Ville's inequality implies:

$$\sup_{(P_X, P_Y) \in \mathcal{P}_0} \mathbb{P}_{P_X \times P_Y}\Big(\sup_{t \geq 1} W_t \geq 1/\alpha\Big) \leq \alpha.$$

Consequently, a sequential test that rejects the null at the stopping time $\tau = \inf\{t \geq 1 : W_t \geq 1/\alpha\}$ satisfies:

$$\sup_{(P_X, P_Y) \in \mathcal{P}_0} \mathbb{P}_{P_X \times P_Y}(\tau < \infty) \leq \alpha.$$

This mathematically guarantees that the probability of a false rejection (Type-I error) is strictly bounded by α, *regardless of when the evaluation is halted*. This uniform Type-I error control is formally established by Shekhar & Ramdas (2023, Thm. 1).

Two critical practical advantages emerge from this framework: *anytime-validity* and *post-hoc significance*. Anytime-validity ensures that an evaluator (or arbitrator) can continuously monitor the data stream and safely reject the null as soon as $W_t \geq 1/\alpha$, completely avoiding Type-I error inflation. In stark contrast, classical fixed-horizon p-values require a rigidly predefined sample size. Repeatedly "peeking" at the data and dynamically stopping when $p \leq \alpha$ catastrophically inflates the False Positive Rate unless complex α-spending corrections are applied.

Furthermore, the accumulated wealth W_t naturally provides *post-hoc significance*. The magnitude of the e-value serves as a valid, interpretable measure of evidence independent of any strictly predefined threshold. If the terminal wealth vastly exceeds the initial target, the arbitrator can legitimately claim a higher degree of statistical significance. Conversely, if W_t falls short of the intended α-level, the arbitrator can still reject the null, albeit at a less stringent, data-dependent level when the e-value is sufficiently large.

Ultimately, these theoretical properties make e-values exceptionally well-suited for Dataset Inference. They offer safe continuous monitoring, flexible post-hoc interpretation, and—because expected log-wealth grows linearly under the alternative hypothesis—exponential evidence accumulation that yields massive gains in sample efficiency.

C.4 Implementation Details

For the Kernel-MMD discrepancy metric, we utilize a degree-2 polynomial kernel:

$$K(x, y) = (\gamma\, x^\top y + c)^2, \qquad \gamma = \tfrac{1}{p},\ c = 1,$$

where p is the feature dimension. This formulation preserves sensitivity to *mean differences* via the linear term (enabled by $c = 1$) which is the primary signal of interest in Dataset Inference while still capturing second-order interactions. To strictly satisfy the theoretical requirement that the discrepancy metric be bounded, we apply a $\tanh$ transformation to the sequential witness function increments $\widehat{h}_t$ to ensure boundedness in $(-1, 1)$.

For both the ONS and Kelly betting strategies, we enforce a maximum betting cap of $\lambda_{\max} = 0.5$, which strictly restricts the wagered stake to the interval $[-\lambda_{\max}, \lambda_{\max}]$. While higher thresholds permit more aggressive betting, this strict bounding is mathematically necessary to completely prevent the wealth process from reaching zero (ruin). Our pipeline guarantees this by explicitly clipping the calculated stakes at the maximum value. Finally, we provide an ablation study in section E.6 to demonstrate that the statistical power of our sequential test remains highly robust to the specific selection of this maximum betting threshold.

We evaluate the sequential test using a Monte Carlo simulation consisting of 500 independent trials. In each replication, a fresh sequence of observations is drawn to simulate the sequential testing process. Accumulating these outcomes across all replications allows us to construct an empirical power curve as well as the empirical distribution of stopping times. To ensure maximum robustness, we repeat this entire simulation across 25 distinct random seeds, reporting the averaged wealth trajectories and stopping times in our final results.

D Detailed Empirical Results

D.1 More Results across Methods and Model Sizes

In fig. 8, we extend the evaluation presented in fig. 2 to include the PhilPapers, Europarl and OpenWebText2 datasets. These extended results further demonstrate that DICE outperforms competing black-box baselines in most subsets, reaching the significance threshold within 2000 data points, and achieves performance remarkably close to DICE+. Specifically, our approach requires significantly fewer observations to cross the strict 0.05 and 0.01 statistical significance thresholds.

D.2 More Results across Model Families

Here we present more results about performance of DICE on different model families namely OLMo-7B and Qwen2-7B model variants.

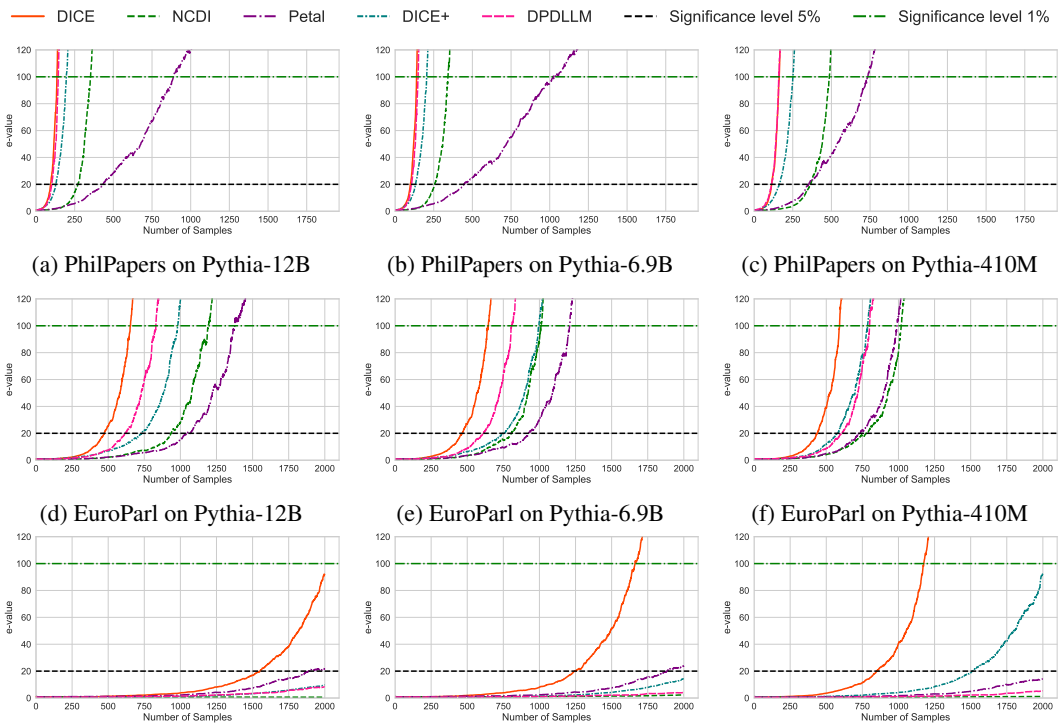
OLMo-7B. We apply the DICE framework to the OLMo-7B model by conducting sequential hypothesis testing via betting. Our evaluation is performed on multiple subsets of the Dolma dataset (Soldaini et al., 2024), and the results are shown in Figure 9 and the detailed performance results are reported in Table 8. Across diverse data sources (including Reddit, StackExchange, Common Crawl, Wikipedia, Gutenberg, and PeS2o) DICE consistently detects membership, with e-values surpassing 100 for most subsets. These findings demonstrate that DICE remains reliable and effective across heterogeneous data distributions and model families.

Qwen2-7B. To further evaluate the robustness of DICE across different model families, we assess our method on both the Qwen2-7B-Base and Qwen2-7B-Instruct (Yang et al., 2024) models using the BookMIA dataset. The results in Figure 10 and Table 7 demonstrate a significant effect in the wealth process, indicating that BookMIA data was indeed included in the training set of Qwen2-7B.

Moreover, our method not only successfully infers membership for the BookMIA dataset, but also remains effective for instruction-tuned models designed for improved alignment. As shown in Figure 10, the instruction-tuned model requires slightly more observations to reach the 1% significance threshold. We hypothesize that this additional difficulty may be due to perturbations introduced into the model's output distribution during the instruction-tuning process.

D.3 Detailed Performance Measures

In this section we provide detailed results of the performance measure for DICE compared to the other baselines. The performance comparison between DICE and the black-box baseline models is



(a) PhilPapers on Pythia-12B (b) PhilPapers on Pythia-6.9B (c) PhilPapers on Pythia-410M

(d) EuroParl on Pythia-12B (e) EuroParl on Pythia-6.9B (f) EuroParl on Pythia-410M

(g) OpenWebText2 on Pythia-12B (h) OpenWebText2 on Pythia-6.9B (i) OpenWebText2 on Pythia-410M

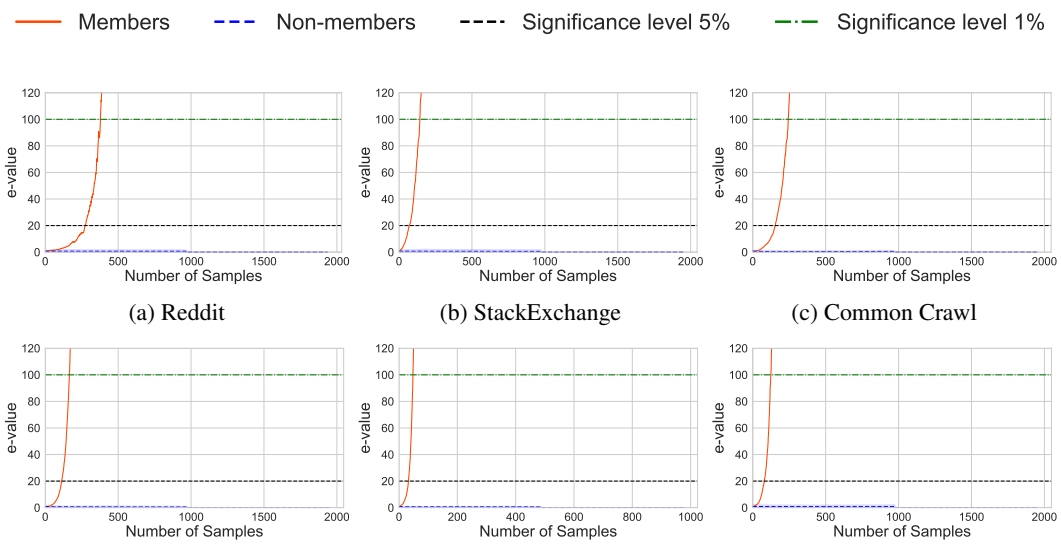
Figure 8: **Comparison across methods and model sizes.** We compare DICE against PETAL, NCDI, DPDLLM and DICE+, reporting accumulated wealth for Pythia-12B (left), Pythia-6.9B (middle), and Pythia-410M (right) on PhilPapers, EuroParl and OpenWebText2 datasets. This is an extension of Figure 2 from the main paper.

detailed in table 3, table 4, and table 5 for the PYTHIA-12B, PYTHIA-6.9B, and PYTHIA-410M architectures, respectively. Additionally, we evaluate DICE against the DICE+ approach on the PYTHIA-12B model, with these results presented in table 6. Across all aforementioned experiments, we consistently utilize Kernel-MMD as the discrepancy metric and Online Newton Step (ONS) as the betting strategy.

A comprehensive evaluation across multiple model scales (PYTHIA-12B, PYTHIA-6.9B, and PYTHIA-410M) demonstrates that DICE outperforms existing black-box baselines in both True Positive Rate (TPR) and sample efficiency for most of the datasets. For instance, on the PYTHIA-12B model, DICE successfully reaches the significance threshold on 17 out of 18 evaluated PILE subsets, whereas competing baselines such as NCDI and PETAL fail to converge (N/A) on up to 8 subsets. Furthermore, DICE consistently achieves near-perfect TPRs (1.000) while requiring a fraction of the observations. On subsets like PhilPapers and Ubuntu IRC, DICE halts after merely 93.8 and 47.0 average observations, respectively, yielding a 2x to 8x efficiency improvement over DPDLLM and PETAL.

Most notably, DICE not only matches but occasionally surpasses the performance of the privileged DICE+ baseline. On subsets where the DICE+ fails to converge entirely (such as FreeLaw and GitHub), DICE maintains robust TPRs of 0.790 and 0.697. Conversely, on highly memorized datasets like Wikipedia and Ubuntu IRC, both methods achieve perfect detection rates with nearly identical stopping times (e.g., 47.0 versus 42.1 observations). These concrete results provide compelling empirical evidence that relying on privileged access to internal model token probabilities is not a necessary condition for executing highly successful and efficient Dataset Inference.

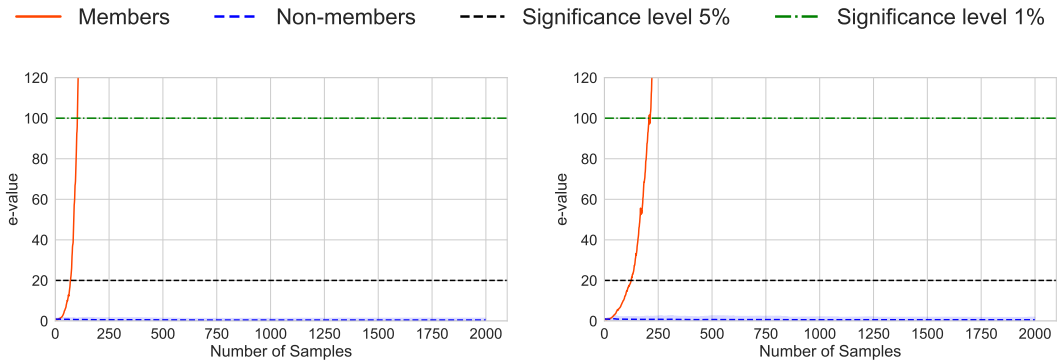
Furthermore, it is notable that the False Positive Rate (FPR) remains strictly controlled across all evaluated methods and datasets. This empirical observation aligns perfectly with the theoretical



(a) Reddit (b) StackExchange (c) Common Crawl

(d) Wikipedia (e) Gutenberg (f) PeS2o

Figure 9: **Testing by betting across DOLMA subsets.** Accumulated wealth with 95% confidence intervals of the e-process performed on various subsets on the OLMo-7B model.



(a) Qwen2-7B-Base (b) Qwen2-7B-Instruct

Figure 10: **DICE for Qwen2 model variants.** The results show that DICE successfully detects the membership of the BookMIA dataset in both models. In addition, the instruction-tuned model requires slightly more observations to reach the 1% significance threshold, suggesting that instruction tuning may introduce perturbations to the output distribution that make membership detection more challenging.

guarantees of type-I error control inherent to sequential testing with e-values, thereby validating the mathematical framework's reliability in practical applications.

E Ablation Studies

E.1 Importance of Extracted Membership Features

To rigorously evaluate the individual contributions of the extracted membership features, we conduct a systematic ablation study. We cumulatively exclude specific feature groups from the pipeline while holding the rest of the evaluation framework constant. We proceed in two stages: first, we remove the Min-$K\%$ Prob and Max-$K\%$ Prob metrics; second, we additionally exclude all perplexity-based features. This strategy explicitly tests our hypothesis that the complete ensemble of proposed features is necessary to achieve optimal detection performance.

Table 3: **Sequential dataset inference performance for Pythia-12B.** True positive rate (TPR), false positive rate (FPR), and average stopping time for testing by betting across Pile subsets. Results compare our method (DICE) against PETAL, NCDI, and DPDLLM. N/A indicates that the test did not reach significance within the budget.

	DICE			PETAL			NCDI			DPDLLM		
Dataset	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop
ArXiv	0.177	0.004	$1893.2_{\pm 21.8}$	0.027	0.004	N/A	0.002	0.009	N/A	0.003	0.002	N/A
BookCorpus2	1.000	0.011	$277.9_{\pm 9.4}$	1.000	0.008	$292.9_{\pm 11.8}$	0.940	0.001	$1128.4_{\pm 31.7}$	0.999	0.004	$277.6_{\pm 10.4}$
Books3	0.999	0.002	$295.8_{\pm 10.4}$	0.994	0.002	$589.9_{\pm 22.5}$	0.002	0.005	N/A	0.998	0.003	$430.5_{\pm 19.9}$
Pile-CC	0.864	0.014	$1360.4_{\pm 38.3}$	0.096	0.008	N/A	0.008	0.030	N/A	0.005	0.004	N/A
EuroParl	0.994	0.009	$504.6_{\pm 17.6}$	0.912	0.015	$1127.4_{\pm 30.4}$	0.971	0.002	$858.2_{\pm 24.6}$	0.993	0.004	$633.6_{\pm 22.2}$
FreeLaw	0.789	0.003	$1445.3_{\pm 39.2}$	0.013	0.002	N/A	0.057	0.002	N/A	0.009	0.002	N/A
GitHub	0.697	0.019	$1555.8_{\pm 36.8}$	0.005	0.002	N/A	0.285	0.002	N/A	0.973	0.003	$860.2_{\pm 35.0}$
Gutenberg	0.985	0.002	$694.3_{\pm 26.7}$	0.794	0.002	$1436.4_{\pm 33.2}$	0.318	0.002	N/A	0.115	0.002	N/A
HackerNews	0.998	0.002	$405.4_{\pm 14.9}$	0.999	0.003	$317.7_{\pm 14.2}$	0.988	0.002	$764.1_{\pm 21.1}$	0.906	0.002	$1174.3_{\pm 36.8}$
DM Mathematics	0.796	0.002	$1473.7_{\pm 36.8}$	0.475	0.007	$1717.3_{\pm 21.6}$	0.002	0.010	N/A	0.016	0.007	N/A
OpenSubtitles	0.008	0.005	N/A	0.004	0.002	N/A	0.003	0.035	N/A	0.947	0.004	$1021.3_{\pm 34.8}$
OpenWebText2	0.759	0.087	$1497.7_{\pm 28.0}$	0.448	0.005	$1735.8_{\pm 19.1}$	0.003	0.001	N/A	0.356	0.077	$1765.7_{\pm 34.8}$
PhilPapers	1.000	0.006	$93.8_{\pm 3.9}$	0.769	0.001	$811.4_{\pm 48.4}$	1.000	0.002	$244.2_{\pm 8.2}$	1.000	0.003	$120.6_{\pm 5.1}$
StackExchange	1.000	0.013	$111.8_{\pm 3.4}$	1.000	0.018	$116.4_{\pm 4.5}$	1.000	0.005	$265.7_{\pm 6.0}$	1.000	0.004	$211.4_{\pm 7.7}$
Ubuntu IRC	1.000	0.006	$47.0_{\pm 1.5}$	1.000	0.003	$106.3_{\pm 4.5}$	1.000	0.012	$94.1_{\pm 3.2}$	1.000	0.003	$121.2_{\pm 5.9}$
USPTO	0.974	0.002	$886.8_{\pm 32.4}$	0.897	0.003	$1211.6_{\pm 37.9}$	0.148	0.007	N/A	0.599	0.003	$1619.3_{\pm 30.4}$
Wikipedia	0.999	0.003	$279.7_{\pm 14.3}$	0.495	0.002	$1704.9_{\pm 34.8}$	0.997	0.002	$533.3_{\pm 17.4}$	0.030	0.004	N/A
YoutubeSubtitles	1.000	0.002	$119.1_{\pm 3.9}$	1.000	0.003	$105.4_{\pm 3.2}$	1.000	0.003	$156.7_{\pm 5.4}$	1.000	0.006	$119.7_{\pm 4.5}$

Table 4: **Sequential dataset inference performance for Pythia-6.9B.** True positive rate (TPR), false positive rate (FPR), and average stopping time for testing by betting across Pile subsets. Results compare our method (DICE) against PETAL, NCDI, and DPDLLM. N/A indicates that the test did not reach significance within the budget.

	DICE			PETAL			NCDI			DPDLLM		
Dataset	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop
ArXiv	0.028	0.003	N/A	0.017	0.002	N/A	0.004	0.002	N/A	0.702	0.007	$1524.4_{\pm 36.1}$
BookCorpus2	1.000	0.003	$281.4_{\pm 8.7}$	1.000	0.025	$294.1_{\pm 10.9}$	0.973	0.004	$948.5_{\pm 31.0}$	1.000	0.006	$265.0_{\pm 10.5}$
Books3	1.000	0.004	$324.2_{\pm 11.3}$	0.992	0.005	$655.1_{\pm 23.0}$	0.003	0.081	N/A	0.998	0.003	$407.5_{\pm 18.6}$
Pile-CC	0.398	0.002	$1765.1_{\pm 30.9}$	0.033	0.003	N/A	0.004	0.003	N/A	0.009	0.004	N/A
EuroParl	0.993	0.002	$516.5_{\pm 19.5}$	0.936	0.006	$1017.4_{\pm 40.1}$	0.972	0.002	$825.9_{\pm 22.5}$	0.995	0.004	$607.9_{\pm 24.4}$
FreeLaw	0.782	0.002	$1467.7_{\pm 42.7}$	0.022	0.003	N/A	0.016	0.008	N/A	0.008	0.005	N/A
GitHub	0.750	0.004	$1496.0_{\pm 38.4}$	0.004	0.004	N/A	0.741	0.002	$1512.0_{\pm 36.9}$	0.962	0.004	$956.9_{\pm 38.0}$
Gutenberg	0.970	0.005	$871.7_{\pm 26.7}$	0.854	0.020	$1325.9_{\pm 31.4}$	0.029	0.002	N/A	0.401	0.003	$1763.7_{\pm 33.3}$
HackerNews	0.997	0.001	$424.8_{\pm 16.8}$	0.998	0.003	$331.1_{\pm 13.7}$	0.990	0.003	$768.8_{\pm 23.8}$	0.861	0.004	$1290.1_{\pm 35.8}$
DM Mathematics	0.879	0.003	$1314.8_{\pm 37.2}$	0.487	0.004	$1713.1_{\pm 32.3}$	0.066	0.002	N/A	0.011	0.003	N/A
OpenSubtitles	0.003	0.003	N/A	0.004	0.005	N/A	0.003	0.002	N/A	0.911	0.009	$1143.6_{\pm 41.0}$
OpenWebText2	0.844	0.059	$1347.6_{\pm 27.6}$	0.453	0.006	$1741.8_{\pm 24.0}$	0.030	0.001	N/A	0.174	0.003	$1868.3_{\pm 23.6}$
PhilPapers	1.000	0.003	$96.8_{\pm 3.8}$	0.738	0.003	$841.5_{\pm 48.6}$	1.000	0.004	$278.6_{\pm 9.6}$	1.000	0.003	$123.4_{\pm 4.0}$
StackExchange	1.000	0.003	$112.2_{\pm 3.9}$	1.000	0.004	$117.9_{\pm 4.8}$	1.000	0.003	$231.1_{\pm 8.8}$	1.000	0.012	$214.3_{\pm 8.0}$
Ubuntu IRC	1.000	0.010	$46.9_{\pm 1.7}$	1.000	0.003	$114.1_{\pm 5.2}$	1.000	0.001	$88.1_{\pm 2.7}$	1.000	0.004	$122.5_{\pm 4.8}$
USPTO	0.972	0.001	$904.8_{\pm 29.6}$	0.910	0.006	$1182.9_{\pm 39.4}$	0.007	0.007	N/A	0.661	0.003	$1585.8_{\pm 29.6}$
Wikipedia	1.000	0.006	$260.0_{\pm 13.6}$	0.598	0.006	$1647.0_{\pm 38.5}$	0.996	0.006	$552.8_{\pm 13.5}$	0.028	0.002	N/A
YoutubeSubtitles	1.000	0.009	$118.2_{\pm 4.5}$	1.000	0.004	$104.4_{\pm 3.1}$	1.000	0.003	$148.3_{\pm 5.8}$	1.000	0.013	$120.7_{\pm 4.6}$

When equipped with the full feature set, DICE achieves an average TPR of 0.836 across the PILE subsets on the PYTHIA-12B model. However, removing the first set of probability metrics reduces the average TPR to 0.775, and the subsequent removal of the perplexity features causes a drastic degradation to 0.597. This performance collapse is particularly pronounced in the BOOKS3, PILE-CC, DM MATHEMATICS, and GUTENBERG subsets, which experience severe TPR drops of 0.995, 0.852, 0.797, and 0.752, respectively. These empirical findings confirm that our comprehensive feature ensemble is essential for effective performance, allowing the pipeline to maintain a high TPR while minimizing the number of observations required to reach the statistical significance threshold.

E.2 ROBUSTNESS TO THE EMBEDDING MODEL

To demonstrate the robustness of our approach to the choice of embedding architecture, we replicate our experiments using the `stella_en_1.5B_v5` model in place of the `all-MiniLM-L6-v2` model utilized in our primary evaluation. This supplementary analysis targets the PYTHIA-12B model on the UBUNTU IRC subset of the PILE dataset.

Utilizing the `stella_en_1.5B_v5` embeddings, our method achieves a perfect True Positive Rate (TPR) of 1.000 and a False Positive Rate (FPR) of 0.0006, requiring an average of only 39.3 ± 1.0

Table 5: **Sequential dataset inference performance for Pythia-410M.** True positive rate (TPR), false positive rate (FPR), and average stopping time for testing by betting across Pile subsets. Results compare our method (DICE) against PETAL, NCDI, and DPDLLM. N/A indicates that the test did not reach significance within the budget.

	DICE			PETAL			NCDI			DPDLLM		
Dataset	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop
ArXiv	0.005	0.002	N/A	0.005	0.009	N/A	0.003	0.002	N/A	0.698	0.008	$1529.2_{\pm 42.2}$
BookCorpus2	0.999	0.003	$449.7_{\pm 12.6}$	0.998	0.003	$397.2_{\pm 14.8}$	0.986	0.002	$840.5_{\pm 20.8}$	0.999	0.007	$282.9_{\pm 11.2}$
Books3	0.996	0.004	$544.9_{\pm 21.4}$	0.978	0.004	$871.5_{\pm 30.6}$	0.072	0.002	N/A	0.999	0.006	$414.5_{\pm 21.6}$
Pile-CC	0.004	0.001	N/A	0.005	0.004	N/A	0.004	0.003	N/A	0.007	0.004	N/A
EuroParl	0.995	0.001	$460.5_{\pm 20.0}$	0.964	0.016	$819.2_{\pm 28.7}$	0.967	0.002	$857.6_{\pm 34.5}$	0.995	0.004	$613.1_{\pm 21.3}$
FreeLaw	0.883	0.005	$1296.8_{\pm 39.6}$	0.048	0.006	N/A	0.054	0.003	N/A	0.004	0.008	N/A
GitHub	0.415	0.060	$1794.8_{\pm 31.5}$	0.005	0.087	N/A	0.250	0.005	N/A	0.976	0.005	$838.1_{\pm 35.6}$
Gutenberg	0.528	0.002	$1722.1_{\pm 31.1}$	0.963	0.004	$957.0_{\pm 36.4}$	0.003	0.098	N/A	0.123	0.004	N/A
HackerNews	0.998	0.002	$432.9_{\pm 17.8}$	0.999	0.056	$363.1_{\pm 15.6}$	0.564	0.001	$1680.0_{\pm 25.9}$	0.825	0.003	$1368.7_{\pm 31.1}$
DM Mathematics	0.383	0.026	$1795.7_{\pm 23.5}$	0.081	0.006	N/A	0.005	0.004	N/A	0.009	0.002	N/A
OpenSubtitles	0.001	0.001	N/A	0.003	0.002	N/A	0.003	0.003	N/A	0.925	0.004	$1125.0_{\pm 40.9}$
OpenWebText2	0.966	0.002	$906.7_{\pm 31.0}$	0.461	0.007	$1735.1_{\pm 27.6}$	0.013	0.004	N/A	0.268	0.003	$1818.3_{\pm 18.6}$
PhilPapers	1.000	0.002	$118.9_{\pm 4.7}$	0.661	0.644	$908.5_{\pm 47.0}$	1.000	0.001	$360.2_{\pm 17.0}$	1.000	0.004	$136.5_{\pm 4.9}$
StackExchange	1.000	0.002	$125.9_{\pm 4.4}$	1.000	0.007	$130.3_{\pm 5.0}$	1.000	0.013	$247.4_{\pm 9.1}$	1.000	0.003	$213.9_{\pm 8.7}$
Ubuntu IRC	1.000	0.016	$50.2_{\pm 1.6}$	1.000	0.002	$241.3_{\pm 12.1}$	1.000	0.005	$113.7_{\pm 3.2}$	1.000	0.005	$130.2_{\pm 6.1}$
USPTO	0.990	0.003	$718.1_{\pm 27.6}$	0.963	0.005	$950.5_{\pm 29.4}$	0.021	0.007	N/A	0.366	0.006	$1758.9_{\pm 33.9}$
Wikipedia	1.000	0.002	$221.4_{\pm 9.2}$	0.461	0.037	$1734.8_{\pm 34.3}$	0.994	0.002	$587.4_{\pm 20.1}$	0.015	0.003	N/A
YoutubeSubtitles	1.000	0.004	$116.6_{\pm 4.7}$	1.000	0.002	$101.7_{\pm 2.9}$	1.000	0.004	$122.9_{\pm 4.4}$	1.000	0.004	$124.7_{\pm 4.2}$

Table 6: **Sequential dataset inference performance for Pythia-12B: DICE vs. DICE+.** True positive rate (TPR), false positive rate (FPR), and average stopping time for testing by betting across Pile subsets. The DICE+ method uses actual per-token probabilities as privileged information. N/A indicates that the test did not reach significance within the budget.

	DICE			DICE+		
Dataset	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop
ArXiv	0.177	0.004	$1893.2_{\pm 21.8}$	0.585	0.011	$1644.4_{\pm 28.4}$
BookCorpus2	1.000	0.011	$278.0_{\pm 9.4}$	1.000	0.007	$253.4_{\pm 10.0}$
Books3	1.000	0.002	$295.8_{\pm 10.4}$	1.000	0.003	$217.8_{\pm 7.7}$
Pile-CC	0.864	0.015	$1360.4_{\pm 38.3}$	0.931	0.003	$1110.5_{\pm 30.9}$
EuroParl	0.994	0.010	$504.6_{\pm 17.6}$	0.970	0.003	$772.0_{\pm 30.5}$
FreeLaw	0.790	0.003	$1445.3_{\pm 39.2}$	0.059	0.004	N/A
GitHub	0.697	0.019	$1555.8_{\pm 36.8}$	0.011	0.019	N/A
Gutenberg	0.985	0.002	$694.3_{\pm 26.7}$	0.976	0.003	$841.6_{\pm 32.3}$
HackerNews	0.998	0.002	$405.4_{\pm 14.9}$	1.000	0.003	$288.2_{\pm 12.3}$
DM Mathematics	0.797	0.002	$1473.7_{\pm 36.8}$	0.769	0.003	$1502.0_{\pm 26.1}$
OpenSubtitles	0.008	0.005	N/A	0.005	0.003	N/A
OpenWebText2	0.759	0.087	$1497.7_{\pm 28.0}$	0.397	0.003	$1807.7_{\pm 26.8}$
PhilPapers	1.000	0.006	$93.9_{\pm 4.0}$	1.000	0.003	$148.9_{\pm 7.0}$
StackExchange	1.000	0.014	$111.8_{\pm 3.4}$	1.000	0.004	$113.1_{\pm 5.5}$
Ubuntu IRC	1.000	0.006	$47.0_{\pm 1.5}$	1.000	0.004	$42.1_{\pm 1.3}$
USPTO	0.974	0.002	$886.8_{\pm 32.4}$	0.948	0.009	$1043.8_{\pm 35.6}$
Wikipedia	1.000	0.003	$279.7_{\pm 14.3}$	1.000	0.004	$311.9_{\pm 13.4}$
YoutubeSubtitles	1.000	0.002	$119.1_{\pm 3.9}$	1.000	0.001	$112.3_{\pm 4.1}$

observations to reach the 5% significance threshold. Comparing these findings to the original UBUNTU IRC results presented in table 3 reveals no significant deviation in detection capabilities or sample efficiency. This outcome strongly confirms our hypothesis: provided the model meaningfully encodes the semantic structure of the input text, the efficacy of our sequential testing framework is fundamentally independent of the specific embedding architecture.

E.3 EFFECTIVENESS OF NORMALIZATION AND TRIMMING

In our pipeline, the extracted membership features undergo normalization and trimming before being passed to the kernel-MMD discrepancy function (as detailed in section A.5). To empirically validate the necessity of these preprocessing steps, we conduct an ablation study comparing the full DICE framework against a baseline variant where normalization and trimming are entirely omitted. We hypothesize that this preprocessing stage is critical for stabilizing the features and achieving robust, effective detection performance.

Table 7: **Sequential dataset inference performance for Qwen2-7B variants.** True positive rate (TPR), false positive rate (FPR), and average stopping time for testing by betting for BookMIA dataset.

	Qwen2-7B			Qwen2-7B-Instruct		
Dataset	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop
BookMIA	0.963	0.010	$241.1_{\pm 100.5}$	0.958	0.009	$261.6_{\pm 105.9}$

Table 8: **Sequential dataset inference performance for OLMo-7B.** True positive rate (TPR), false positive rate (FPR), and average stopping time for testing by betting.

	OLMo-7B		
Dataset	TPR	FPR	Avg. stop
Common Crawl	0.999	0.007	$198.1_{\pm 9.5}$
Gutenberg	1.000	0.002	$31.4_{\pm 1.1}$
PeS2o	1.000	0.002	$94.6_{\pm 3.5}$
Reddit	0.999	0.004	$362.5_{\pm 14.2}$
Stack Exchange	0.985	0.007	$282.7_{\pm 22.2}$
Wikipedia	0.999	0.004	$123.5_{\pm 8.5}$

The empirical results strongly confirm this hypothesis. Omitting normalization and trimming causes the average True Positive Rate (TPR) across the entire PILE dataset to plummet from 0.836 to 0.506, representing a substantial degradation in performance. Furthermore, this omission severely impacts efficiency. Without these preprocessing steps, the test fails to converge within the 2000 sample limit for several subsets, including PILE-CC, FREELAW, GITHUB, OPENWEBTEXT2, and USPTO. Even among the subsets that successfully reach the 5% significance threshold, the average stopping time across the overall PILE dataset increases from 761.3 to 802.4 observations. These findings definitively demonstrate that normalization and trimming are essential for yielding reliable test outcomes and ensuring rapid convergence with minimal data.

E.4 ROBUSTNESS TO POLLUTED HELD-OUT SET (VALIDATION DATA)

To evaluate the robustness of our sequential testing procedure under held-out-set (validation data) contamination, we conduct an experiment in which a specified proportion of non–members is replaced with true training members. We then apply our hypothesis testing by betting method to the Ubuntu subset of the PILE dataset using the PYTHIA-12B model.

In this framework, the bettor adaptively places stakes based on all observations seen so far. Under the alternative hypothesis, accumulated evidence causes the wealth process to grow. We execute the experiment over 25 random seeds and report the average wealth trajectory in Figure 11. For pollution levels up to 70%, the test reliably gathers sufficient evidence to cross the 1% significance threshold within the first 200 observations. At 80% pollution, the rate of wealth growth diminishes, but the trajectory still shows consistent upward movement, indicating that the remaining clean examples provide enough evidence against the null. At 90% pollution, the curves flatten substantially, reflecting that the contaminated distribution becomes nearly indistinguishable from the null.

Overall, the results demonstrate that the betting-based sequential test is robust to substantial contamination of the held-out set (validation data). Even with as much as 70% of the non–members replaced by members, the procedure continues to accumulate evidence and successfully rejects the null hypothesis.

A notable aspect of Figure 11 is that the curve for a 30% pollution ratio reaches the 1% significance threshold with fewer observations than the 20% curve, and even performs comparably to the clean (0%) baseline. At first glance, this appears counter–intuitive, since higher contamination should make the hypothesis test harder. However, this behavior is fully consistent with the properties of the wealth process. The e–value process is a stochastic, path–dependent quantity. In our experiments, each curve represents an average over 25 random seeds, where the order of samples is independently shuffled at each run. Because the sequential test operates online, observing data points one at a time, early

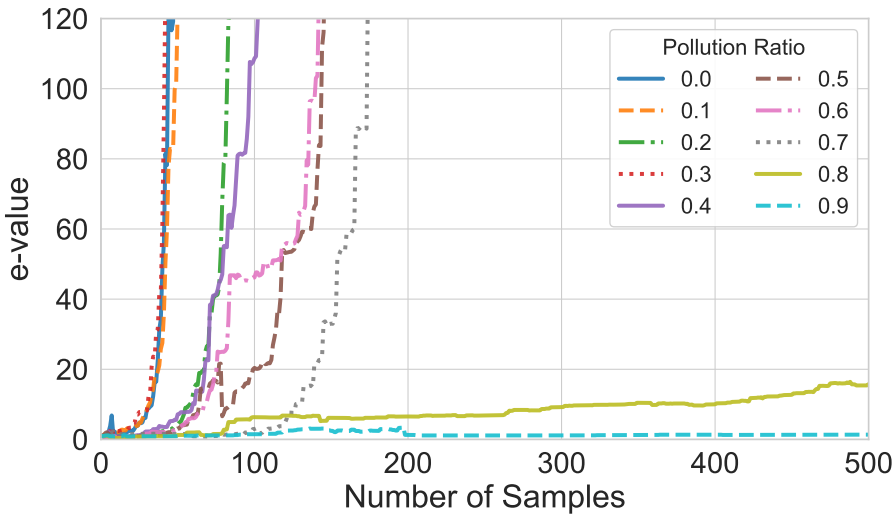


Figure 11: **Effect of pollution ratio in the held-out set on sequential e-value growth.** The plot shows the mean wealth (e-value) trajectories for different non-member pollution ratios ranging from 0 to 0.9. Each curve corresponds to a different injected pollution level. The curves show that once sufficient evidence accumulates against the null hypothesis, the wealth process increases and eventually crosses the significance threshold.

observations with a large discrepancy between the null and alternative can yield disproportionately large multiplicative jumps in the wealth. Consequently, stochasticity can occasionally introduce early data points that appear highly informative to the bettor, causing faster boundary crossing even when the overall setting is less favorable.

Importantly, theory does not require that a "stronger" or cleaner alternative always leads to earlier stopping. For e–processes, the only guarantee is that under the alternative the wealth will eventually grow and cross the threshold with high probability; it does not impose monotonicity of stopping times across different alternatives or contamination levels. The variability observed in Figure 11 therefore reflects inherent stochasticity in the sequential evidence accumulation process, rather than a violation of the expected theoretical behavior.

E.5 Ablation on Betting Strategy

Here, we present ablation results for DICE, evaluated on the PYTHIA-12B model across PILE subsets. The primary goal of this study is to demonstrate that our detection capabilities remain robust to the selection of the underlying betting strategy. To this end, we compare the performance of the two strategies discussed in section C.2: Online Newton Step (ONS) and Kelly Betting. We emphasize that betting strategies serve as modular structural choices within our framework, which flexibly accommodates various configurations. This ablation confirms that our strong performance is not merely an artifact of selecting one specific setup. As evidenced by the results in table 9, both betting strategies yield consistently similar and highly effective outcomes.

E.6 Ablation on Test Power

In this subsection, we examine how the choice of betting strategy and associated hyperparameters influences the statistical power of hypothesis testing by betting. In particular, we show that the testing power is largely robust to both the selection of the betting strategy and the choice of the maximum betting cap. Finally, we compare the sequential testing power of DICE with that of classical hypothesis tests based on fixed-sample p-values.

To quantify this robustness, we compute the *sequential power curve* across repeated Monte Carlo simulations. The sequential power at time t is defined as the probability that the test has rejected the null hypothesis on or before time t. Formally, for a given t, let T_i denote the stopping time for the

Table 9: **Ablation on betting strategies for Pythia-12B.** Comparison of approximate Kelly betting and ONS betting strategies for sequential dataset inference across Pile subsets. N/A indicates that the test did not reach significance within the budget.

Dataset	Betting Strategy					
	ONS Betting			Kelly Betting		
	TPR	FPR	Avg. stop	TPR	FPR	Avg. stop
ArXiv	0.177	0.004	$1893.2_{\pm 21.8}$	0.090	0.002	N/A
BookCorpus2	1.000	0.011	$277.9_{\pm 9.4}$	1.000	0.002	$268.2_{\pm 9.2}$
Books3	0.999	0.002	$295.8_{\pm 10.4}$	0.999	0.007	$310.4_{\pm 11.1}$
Pile-CC	0.864	0.014	$1360.4_{\pm 38.3}$	0.799	0.009	$1479.1_{\pm 38.7}$
EuroParl	0.994	0.009	$504.6_{\pm 17.6}$	0.987	0.002	$602.3_{\pm 21.6}$
FreeLaw	0.789	0.003	$1445.3_{\pm 39.2}$	0.651	0.002	$1614.2_{\pm 35.01}$
GitHub	0.697	0.019	$1555.8_{\pm 36.8}$	0.708	0.002	$1550.2_{\pm 35.5}$
Gutenberg	0.985	0.002	$694.3_{\pm 26.7}$	0.968	0.001	$851.8_{\pm 31.4}$
HackerNews	0.998	0.002	$405.4_{\pm 14.9}$	0.998	0.004	$401.5_{\pm 14.5}$
DM Mathematics	0.796	0.002	$1473.7_{\pm 36.8}$	0.926	0.001	$1214.2_{\pm 32.8}$
OpenSubtitles	0.008	0.005	N/A	0.005	0.002	N/A
OpenWebText2	0.759	0.087	$1497.7_{\pm 28.0}$	0.667	0.002	$1608.2_{\pm 27.5}$
PhilPapers	1.000	0.006	$93.8_{\pm 3.9}$	1.000	0.002	$109.4_{\pm 4.4}$
StackExchange	1.000	0.013	$111.8_{\pm 3.4}$	1.000	0.003	$125.9_{\pm 4.2}$
Ubuntu IRC	1.000	0.006	$47.0_{\pm 1.5}$	1.000	0.002	$45.8_{\pm 1.3}$
USPTO	0.974	0.002	$886.8_{\pm 32.4}$	0.973	0.002	$916.6_{\pm 34.2}$
Wikipedia	0.999	0.003	$279.7_{\pm 14.3}$	0.999	0.003	$282.8_{\pm 14.0}$
YoutubeSubtitles	1.000	0.002	$119.1_{\pm 3.9}$	1.000	0.017	$118.4_{\pm 3.9}$

i-th trial. The empirical sequential power is then estimated as:

$$\text{Power}(t) = \frac{1}{\text{num. trials}} \sum_{i=1}^{\text{num. trials}} \mathbf{1}\{T_i \leq t\}.$$

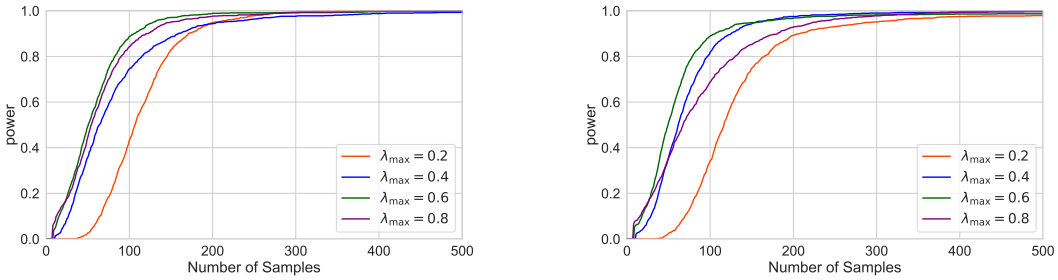
As shown in Figure 12, we plot the sequential power curves for the Ubuntu subset of the PILE dataset using MI features extracted by DICE on PYTHIA-12B model. The results demonstrate that the hypothesis test is robust to both the choice of betting strategy and the maximum betting cap. In accordance with log-optimality constraints, the maximum betting value must lie within the interval $[-1, 1]$ to prevent the wealth process from collapsing to zero. Within this feasible range, both the ONS and approximate Kelly strategies achieve their highest empirical power at a maximum betting value of 0.6. However, the sequential power remains stable across different cap values, consistently converging to the significance threshold under the alternative hypothesis.

We additionally compare the power of the sequential testing procedure described above with that of a classical fixed-sample permutation test. As in the sequential setting, we consider aggregated membership scores for the suspect and held-out sets, denoted by X and Y, respectively, and use the same kernel maximum mean discrepancy (MMD) with a polynomial kernel as the measure of distributional distance.

A classical *two-sample permutation test with kernel MMD* assesses whether two collections of samples, $X \sim P$ and $Y \sim Q$, are drawn from the same distribution ($H_0 : P = Q$). The test uses the *maximum mean discrepancy (MMD)* as a discrepancy statistic and approximates its null distribution via *label permutations*. We compute an observed test statistic $T_{\text{obs}} = \widehat{\text{MMD}}^2(X, Y)$ that increases when $P \neq Q$ using the *biased squared MMD* estimator:

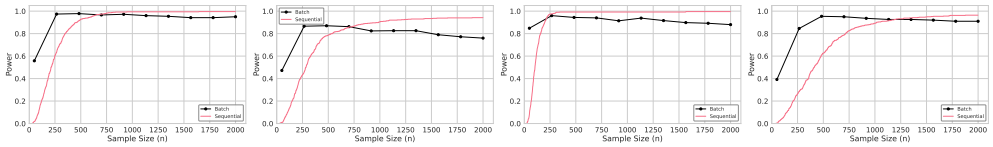
$$\widehat{\text{MMD}}^2(X, Y) = \frac{1}{n^2} \sum_{i,i'} k(x_i, x_{i'}) + \frac{1}{n^2} \sum_{j,j'} k(y_j, y_{j'}) - \frac{2}{n^2} \sum_{i,j} k(x_i, y_j).$$

Under the null hypothesis, the pooled sample $Z = X \cup Y$ is *exchangeable* with respect to the labels indicating membership in X or Y. To approximate the null distribution of the test statistic, we repeatedly (with $B = 200$ permutations in our experiments) draw a random permutation of the pooled indices, split the permuted samples into two groups of equal size n to form pseudo-samples $(X^{(b)}, Y^{(b)})$, and recompute the squared MMD statistic $T^{(b)} = \widehat{\text{MMD}}^2\big(X^{(b)}, Y^{(b)}\big)$. The



(a) Sequential Power Curve (Kelly Betting). (b) Sequential Power Curve (ONS Betting).

Figure 12: **Robustness of sequential power across betting strategies.** We compare sequential power curves for the approximate Kelly strategy and the ONS strategy on the Ubuntu subset. Results show that detection performance is consistent across strategies.



(a) Wikipedia (b) EuroParl (c) StackExchange (d) Gutenberg

Figure 13: **Statistical power comparison.** We compare sequential power curves for the sequential hypothesis testing by betting to that of classical two-sample permutation test with kernel-MMD.

permutation p-value is then defined as the fraction of permuted statistics at least as large as the observed value,

$$\hat{p} = \frac{1 + \sum_{b=1}^{B} \mathbf{1}\{T^{(b)} \geq T_{\text{obs}}\}}{B + 1},$$

corresponding to a one-sided test in which larger MMD values provide stronger evidence against H_0. The null hypothesis is rejected at significance level α if $\hat{p} \leq \alpha$.

This testing procedure is fully nonparametric and relies solely on exchangeability under the null hypothesis. With a sufficient number of permutations, it provides an accurate approximation to the null distribution of the MMD statistic. To estimate statistical power, we repeat the entire procedure across multiple Monte Carlo trials (we use 20 trials) and over increasing sample sizes. The empirical power at a given sample size n is then defined as the fraction of trials in which the permutation p-value falls below the chosen significance level α.

The results of this comparison are shown in Figure 13, which reports empirical power as a function of sample size for both the sequential betting-based test and the classical permutation test. The permutation test is evaluated only at predetermined sample sizes and therefore exhibits a delayed increase in power, reflecting its fixed-sample nature. In contrast, the sequential test accumulates evidence continuously over observations, leading to a smooth and monotonic increase in power as more samples are observed. Across all datasets, the sequential procedure consistently reaches high power, and in particular power close to one, at substantially smaller sample sizes than the permutation test. This demonstrates that sequential testing enables earlier and more sample-efficient rejection of the null hypothesis while maintaining valid Type-I error control under optional stopping, whereas the classical permutation test is only valid when evaluated at a fixed, pre-specified sample size.

E.7 Robustness to Non-member Selection

To further evaluate the robustness of the hypothesis testing by betting framework with respect to the selection of non-member data points, we used the MI scores extracted by our method from the Pythia-12B model on the Ubuntu IRC subset of The Pile dataset. We randomly shuffled the data and divided it into batches of 100 data points each. We then applied the sequential test independently to every batch. The full dataset contains 2,000 member and non-member data points, resulting in 20 subsets of 100 points each. For every subset, we ran the test with 50 random seeds and plotted the average log wealth across seeds.

The results are shown in Figure 14. In every batch, the test successfully accumulates sufficient evidence to reject the null hypothesis and reaches the 1% significance threshold. This demonstrates that the test is robust to the choice of non-member samples: as long as the discrepancy metric provides enough evidence, the betting strategy allocates wagers so that the wealth grows exponentially under the alternative.

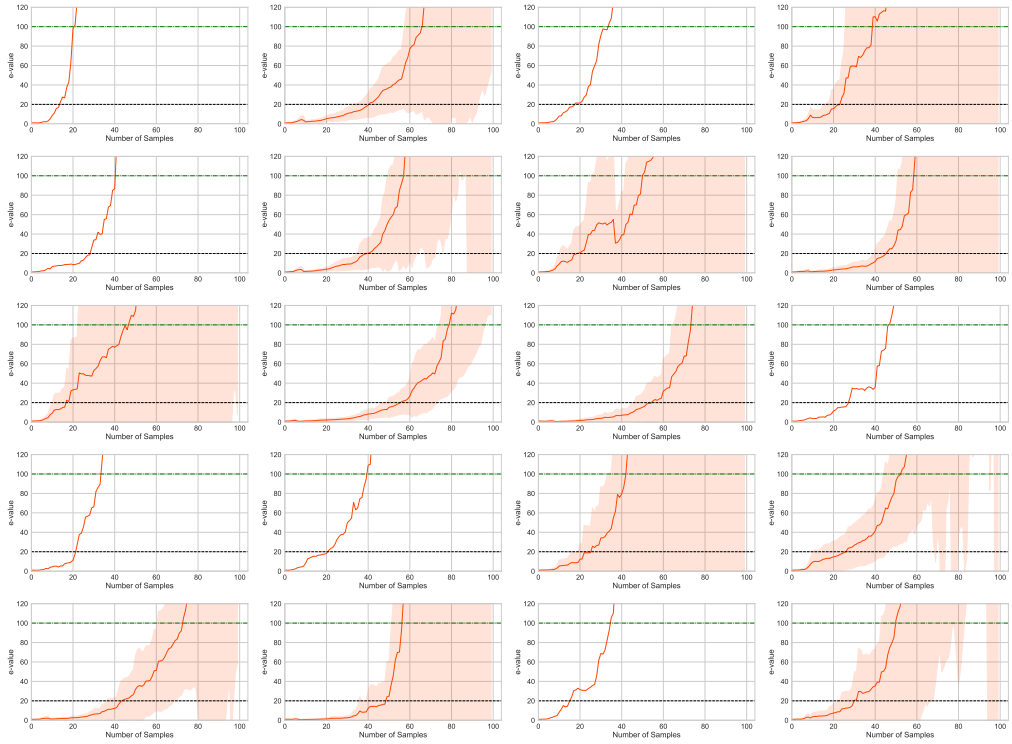


Figure 14: **E-process trajectories for 20 batches of 100 data points from the Ubuntu IRC subset of the Pile dataset**. We use MI features collected with our method from the Pythia-12B model. The results show that the betting-based testing framework is robust in distinguishing non-member data points, consistently accumulating sufficient evidence to reject the null hypothesis in each subset independently.

E.8 Black-Box API Cost Analysis

On the WikiMIA-24 dataset, our sequential test achieves statistical significance after processing only 264 samples.

We rely on two shared query types to extract all necessary token-level and sequence-level membership features:

- **Token-Level Feature Extraction:** This process requires 127 sequential calls per sample, with each call generating a single output token. Across the 264 evaluated samples, this amounts to 33,528 API calls, consuming approximately 2.15 million input tokens and 33.5 thousand output tokens.
- **Sequence-Level Suffix Generation:** Applying multiple perturbations (as detailed in section A.3) requires 7 calls per sample: one for the original sequence and six for the structural variants. Each call processes a 96-token prefix and generates a 32-token output. Across 264 samples, this adds 1,848 API calls, corresponding to roughly 177,400 input tokens and 59,100 output tokens.

In total, reaching the sequential stopping point necessitates 35,376 API calls, translating to approximately 2.32 million input tokens and 92,600 output tokens. Based on standard GPT-5 pricing ($1.25 per 1M input tokens and $10.00 per 1M output tokens), successfully detecting dataset membership costs a mere $3.83. If executed via the OpenAI Batch API, this cost drops to just $1.92.

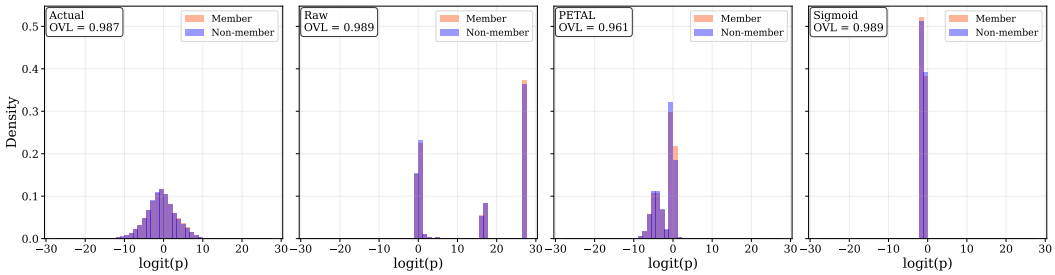


Figure 15: **Per-token logit distributions.** This figure shows histograms of per-token probabilities expressed in logit space for member and non-member samples. We quantify the separability of these distributions using the overlap (OVL) metric, which measures the degree of overlap between the two histograms. The OVL value ranges from 0 to 1, where 1 indicates complete overlap and lower values indicate greater distributional separation.

Crucially, extracting our comprehensive suite of 27 features does not require 27 independent sets of queries. Because all features are efficiently derived from the two shared query types outlined above, the marginal cost of achieving the superior adaptive performance of DICE is practically negligible.

F Sigmoid-based Token Probability Estimation

We investigate the underlying factors contributing to the strong performance of sigmoid-based token probability estimation and seek to explain why, this black-box approach performs on par with DICE+ on certain subsets of the Pile dataset while outperforming other black-box baselines.

To this end, we extract the ground-truth per-token log-probabilities from model generations and compare them against the per-token probability estimates produced by the Raw, PETAL, and Sigmoid methods. To enable an unconstrained and numerically stable comparison, we map all per-token probabilities to logit space, yielding values in the range $(-\infty, +\infty)$. We then analyze the resulting distributions by plotting density histograms of per-token logit values for member and non-member samples, as shown in Figure 15. The degree of separability between these distributions is quantified using an overlap (OVL) metric.

In addition, to directly assess the accuracy of probability estimation, we compute the mean absolute error (MAE) between the estimated and ground-truth per-token logits. All experiments are conducted using the Pythia-12B model across 18 subsets of the Pile dataset, with the MAE results summarized in Table 10.

From Figure 15, we observe that PETAL achieves the strongest separation between member and non-member samples, which is consistent with its low mean absolute error (MAE) in per-token probability estimation. However, in the final dataset inference results, DICE consistently outperforms both PETAL and the Raw across nearly all Pile subsets (Table 3) and often exhibits evidence accumulation rate similar to that of DICE+ (Figure 8).

This discrepancy can be attributed to how sigmoid-based calibration estimates per-token probabilities. As shown in Figure 15, the true per-token logits follow a heavy-tailed distribution, a behavior that is also captured by PETAL and is further amplified in the Raw method, which produces numerous extreme outlier estimates. These heavy tails adversely affect downstream statistical testing and largely explain the weaker performance of likelihood-based baselines. In contrast, although sigmoid calibration yields higher estimation error and greater overlap between member and non-member distributions, it preserves the correct directional signal while avoiding heavy-tailed behavior. This property explains why the final method has an on-par performance with DICE+. Furthermore, the rapid accumulation of evidence against the null hypothesis in our approach is largely driven by the use of a diverse, multidimensional feature set. By leveraging these multiple features simultaneously, our framework achieves a more pronounced distributional discrepancy. This amplified signal naturally dictates higher betting stakes within the sequential pipeline, directly accelerating the exponential growth of wealth compared to other baseline methods.

Table 10: **Per-token probability estimation error.** Overall Mean MAE for estimation of per-token probabilities in logit space across Pile subsets.

Subset	Sigmoid	PETAL	Raw
ArXiv	3.75	2.67	15.28
BookCorpus2	3.28	1.98	13.25
Books3	3.52	2.21	13.49
Pile-CC	3.62	2.33	13.43
EuroParl	4.33	3.57	14.44
FreeLaw	3.95	2.96	14.92
GitHub	6.29	5.78	16.62
Gutenberg	3.41	2.34	13.60
HackerNews	3.96	2.93	13.40
DM Mathematics	5.22	4.30	13.80
OpenSubtitles	3.46	2.15	14.35
OpenWebText2	3.59	2.33	13.55
PhilPapers	3.78	2.66	13.43
StackExchange	4.40	3.21	15.90
Ubuntu IRC	4.54	3.65	14.77
USPTO	3.42	2.22	14.43
Wikipedia	3.70	2.49	14.30
YoutubeSubtitles	4.68	3.65	14.84

Furthermore, we adopt sigmoid-based mapping to improve numerical stability during feature normalization and downstream statistical testing. In preliminary experiments, likelihood-based features exhibited a large dynamic range, with negative log-probabilities spanning approximately $[0, 15]$. Under high uncertainty, moderate increases in log-loss (e.g., ≈ 10) resulted in extremely large perplexity values, which dominated variance estimates during normalization and collapsed other features toward zero. By contrast, the sigmoid transformation yields bounded, well-scaled features, preventing individual samples from disproportionately influencing normalization. This stabilization also mitigates issues arising from reference-model calibration, as used in PETAL, where confidence mismatches between models can amplify spurious signals. Overall, sigmoid-based calibration provides a numerically robust alternative that retains discriminative information while avoiding the pathological effects of heavy-tailed likelihood-based features.

G Additional Related Work

Membership Inference (MI) and, at an aggregated level, Dataset Inference (DI), require well-calibrated membership features. This calibration ensures that the scale of the assigned score accurately reflects the membership status of a given data point or dataset. Traditional MI methods used to estimate the training data proportion of a language model, such as the one proposed by Tong et al. (2025), often rely on shadow models. Consequently, approaches like RMIA (Zarifzadeh et al., 2024) utilize these shadow models to perform inference in a calibrated manner. However, training shadow models is computationally prohibitive for state-of-the-art Large Language Models (LLMs) containing billions of parameters. Furthermore, modern LLMs are frequently deployed in black-box settings with API access limited strictly to text generation, rendering traditional shadow-model techniques impractical for real-world scenarios. To overcome this, standard DI frameworks (Maini et al., 2024) typically assume access to a held-out dataset sampled from the same underlying distribution as the target model's training data to perform a calibrated inference.

This assumption does not inherently limit the practicality of DI, as recent research has developed robust methodologies for generating synthetic held-out sets. For instance, Zhao et al. (2025) introduce a two-stage generation and calibration process. Initially, a language model is fine-tuned on a text completion task using the suspect set. Subsequently, a classifier-based calibration step is implemented to ensure that any detected signal originates from actual membership rather than distributional shifts. Alternatively, Rossi et al. (2026) leverage the potential existence of natural identifiers (NIDs) within

the suspect set to synthesize generated identifiers (GIDs). These GIDs are sampled via a random generator to construct well-structured strings, such as cryptographic hashes or shortened URLs, that frequently occur in common LLM training corpora. Supported by these advancements in held-out set generation, Dataset Inference remains a highly practical and effective tool for auditing black-box models deployed behind restrictive APIs.