Context-Aware Membership Inference Attacks against Pre-trained Large Language Models

Hongyan Chang[1*] **Ali Shahin Shamsabadi**[2] **Kleomenis Katevas**[2]
Hamed Haddadi[2,3] **Reza Shokri**[4]

[1]Mohamed bin Zayed University of Artificial Intelligence [2]Brave Software
[3]Imperial College London [4]National University of Singapore

arXiv:2409.13745v2 [cs.CL] 16 Sep 2025

Abstract

Membership Inference Attacks (MIAs) on pre-trained Large Language Models (LLMs) aim at determining if a data point was part of the model's training set. Prior MIAs that are built for classification models fail at LLMs, due to ignoring the generative nature of LLMs across token sequences. In this paper, we present a novel attack on pre-trained LLMs that adapts MIA statistical tests to the perplexity dynamics of subsequences within a data point. Our method significantly outperforms prior approaches, revealing context-dependent memorization patterns in pre-trained LLMs.

1 Introduction

To assess memorization and information leakage in models, Membership Inference Attacks (MIAs) aim to determine if a data point was part of a model's training set (Shokri et al., 2017). However, MIAs designed for pre-trained Large Language Models (LLMs) have been largely ineffective (Duan et al., 2024; Das et al., 2024).

This is primarily because these MIAs, originally developed for classification models, fail to account for the generative nature of LLMs. Unlike classification models, which produce a single prediction based on the input, LLMs generate texts token-by-token, adjusting the prediction for each output token based on the *context* of preceding tokens (i.e., the prefix). Prior MIAs overlook this *token-level loss dynamics* and the *influence of prefixes* on the predicted token, both of which contribute to the memorization behaviors of LLMs. As such simplifications miss the critical behaviors of LLMs, notably *context-dependent memorization*, these attacks are often ineffective at identifying training set members in pre-trained LLMs.

Additionally, state-of-the-art MIAs (Zarifzadeh et al., 2024; Carlini et al., 2022; Ye et al., 2022; Mireshghallah et al., 2022) rely on reference models trained similarly to the target model but on a distinct but similarly distributed dataset. Obtaining such reference models is extremely costly and often impractical for pre-trained LLMs. On the other hand, using other available pre-trained models as reference models may also lead to inaccurate attacks due to significant differences in their training processes and model architectures (as is shown analytically (Murakonda et al., 2021) and empirically (Duan et al., 2024) in the literature).

To design a strong MIA against pre-trained LLMs, we need to fully understand how and why memorization occurs during the training. Any piece of text is modeled as a sequence of tokens, and LLMs are trained to maximize the conditional probabilities of generating each token based on the preceding context (i.e., the prefix), by adjusting the model parameters. This process is progressive, as the model adjusts its predictions with each new token, refining its understanding of the sequence.

Key attack insight. Our insight is that memorization is context-dependent, triggered primarily when the prefix provides insufficient information for accurate next-token prediction. If a prefix clearly constrains the possible next tokens, either because it contains repetitive patterns or the next tokens overlap strongly with prefix content, the model can reliably predict the next token through generalization, without significant memorization. In contrast, when the prefix is ambiguous or complex, failing to clearly narrow down subsequent possibilities, the model becomes uncertain. To resolve this uncertainty, the model is more likely to rely on specific memorized sequences encountered during training. Therefore, rather than simply relying on the overall loss across a text sequence as in prior work, an effective MIA must account explicitly for how context influences the model's predictive uncertainty at the token level.

*The work was completed during Hongyan's internship at Brave.

Motivated by this insight, we propose *CAMIA*, a Context-Aware Membership Inference Attack specifically designed to exploit the relationship between prefix ambiguity and memorization. The core idea behind *CAMIA* is straightforward yet powerful: it analyzes how quickly and stably the model transitions from initial uncertainty (high ambiguity) to confident predictions as it generates tokens. By capturing the rate at which prediction uncertainty is resolved, as well as correcting for scenarios where ambiguity is artificially reduced by repetitive content, our method effectively distinguishes memorized sequences from generalized predictions. Unlike prior attacks that rely on static thresholds for average prediction losses, *CAMIA* dynamically adapts its inference strategy at the token-level, directly leveraging the context-dependent nature of memorization to significantly improve inference accuracy.

We provide a comprehensive evaluation of our *CAMIA* on a wide spectrum of pre-trained LLMs from the Pythia (Biderman et al., 2023) and GPT-Neo (Black et al., 2021) suites against prior attacks on the MIMIR benchmark (Duan et al., 2024). The performance increase of our attack is consistent across models of various sizes and 6 data domains. For instance, when attacking the 2.8B Pythia model on member/non-member data sampled from the Arxiv domain, *CAMIA* successfully identifies almost twice more members than prior baselines, increasing the true positive rate from 20.11% to 32% while maintaining a 1% false-positive error rate. [1]

2 Problem Formulation

2.1 Autoregressive language model training

Let $\mathcal{M}$ be an auto-regressive model trained on a private dataset PrivSet $= \{\mathbf{X}_i\}_{i=1}^N$ of size N. Each text $\mathbf{X}_i$ is tokenized into T tokens via a token embedding function, forming a sequence $\{x_1, \ldots, x_T\}$ over a vocabulary $\mathbf{V}$. Let $\mathbf{x}_{<t} = \{x_1, \ldots, x_{t-1}\}$ be the prefix of length $t-1$. The model $\mathcal{M}$ predicts x_t conditioned on $\mathbf{x}_{<t}$, and the prediction loss is defined as the cross-entropy between the predicted distribution $P(x|\mathbf{x}_{<t}; \mathcal{M})$ and the true next token x_t:

$$\mathcal{L}_t(x_t) = -\log P(x_t|\mathbf{x}_{<t}; \mathcal{M}). \quad (1)$$

[1]The code is available in https://github.com/changhongyan123/context_aware_mia

The model minimizes the average next-token loss: $-\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_t(x_t)$.

2.2 Membership inference attack (MIA)

MIAs aim to determine whether a target data point $\mathbf{X}$ was part of the training set PrivSet of a model $\mathcal{M}$. MIAs can be formulated as hypothesis tests: the null hypothesis assumes $\mathbf{X}$ is a non-member, while the alternative assumes it is a member. The adversary's goal is to decide between the two, incurring false positives (non-members misclassified as members) and false negatives (members misclassified as non-members). Following prior work (Yeom et al., 2018; Shi et al., 2023; Zhang et al., 2024a; Carlini et al., 2021), MIAs typically define a membership score $f(\mathbf{X}; \mathcal{M})$ and compare it to a threshold τ to determine membership.

Average loss. A basic approach computes the average next-token loss, $-\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_t(x_t)$, and classifies $\mathbf{X}$ as a member if the score is below τ (Yeom et al., 2018).

Outlier token loss. Min-K% (Shi et al., 2023) averages the losses over the $k\%$ least likely tokens (i.e., with highest $\mathcal{L}_t$), under the intuition that non-members contain more high-loss outliers. Min-K%++ (Zhang et al., 2024a) normalizes each selected token's loss using the expectation and variance of log-probabilities at its position.

Loss calibration. Zlib (Carlini et al., 2021) calibrates the loss by dividing by the input's zlib entropy (Deutsch and Gailly, 1996), i.e., $\mathcal{L}(\mathbf{X}; \mathcal{M})/\text{zlib}(\mathbf{X})$. Reference-based MIA (Carlini et al., 2021) compares losses from $\mathcal{M}$ and a reference model $\mathcal{M}_{\text{ref}}$ via $\mathcal{L}(\mathbf{X}; \mathcal{M}) - \mathcal{L}(\mathbf{X}; \mathcal{M}_{\text{ref}})$, aiming to isolate training-specific memorization. Neighborhood MIA (Mattern et al., 2023) subtracts the average loss over neighbors $\mathcal{N}(\mathbf{X})$ from the loss on $\mathbf{X}$: $\mathcal{L}(\mathbf{X}; \mathcal{M}) - \frac{1}{|\mathcal{N}(\mathbf{X})|}\sum_{\tilde{\mathbf{X}} \in \mathcal{N}(\mathbf{X})}\mathcal{L}(\tilde{\mathbf{X}}; \mathcal{M})$.

2.3 True leakage vs. MIA effectiveness

The effectiveness of Membership Inference Attacks (MIAs) depends primarily on two factors: the model's true leakage (memorization tendency) and the design of the attack algorithm. Models with limited memorization behave similarly for members and non-members, making MIAs inherently challenging (Ye et al., 2022; Carlini et al., 2022). Attack design also critically influences performance; poorly optimized attacks may inaccurately infer membership from prediction correct-

ness alone, failing to account for data and model-specific nuances (Yeom et al., 2018; Carlini et al., 2022). Such challenges motivate our context-aware MIA approach, which explicitly considers context-dependent model behaviors.

Moreover, empirical evaluations of MIAs must carefully handle textual overlaps and dataset construction. Defining membership clearly is particularly difficult for textual data, as minor differences (e.g., punctuation) can obscure exact matches, and overlapping substrings create ambiguity between members and non-members (Duan et al., 2024). Benchmarks that artificially distinguish members from non-members based on external factors can inflate measured attack performance. E.g., WikiMIA separates groups by publication date (Shi et al., 2023) and a *blind baseline*—which predicts membership solely from text content without any model access—already achieves 98.7% AUC, indicating that such benchmarks may not accurately reflect genuine model memorization. *Commercial LLMs* such as GPT-4 (Achiam et al., 2023) often do not disclose training datasets, complicating rigorous evaluations of memorization. To address these evaluation challenges, we adopt the carefully designed MIMIR benchmark (Duan et al., 2024) and focus our experiments on open-source models (e.g., Pythia (Biderman et al., 2023) and GPT-Neo (Black et al., 2021)), ensuring transparent and accurate assessment of MIA effectiveness.

2.4 Threat model and our goal

We adopt the practical threat model from prior work (Shi et al., 2023; Zhang et al., 2024a), where the adversary queries the target LLM $\mathcal{M}$ with arbitrary token sequences and obtains per-token losses (Equation 1), without direct access to $\mathcal{M}$'s architecture or parameters. Practically, per-token loss information was directly accessible via certain APIs (e.g., OpenAI's API before Oct. 2023). Even without direct per-token access, token-level losses can be derived from total sequence losses by comparing incremental predictions (e.g., querying losses for `"The"` and `"The sky"`) (Yeom et al., 2018; Carlini et al., 2021).

Under this practical threat model, our primary goal is to enhance MIAs specifically by leveraging the observation that LLMs memorize training data in a context-dependent manner—memorization is particularly pronounced when the prefix does not sufficiently constrain possible next tokens. Prior MIAs typically aggregate losses at the sequence level and thus fail to exploit these finer-grained, token-level memorization patterns.

To address this limitation, we propose *CAMIA*, a context-aware MIA framework that explicitly captures how prefixes influence the model's reliance on memorization. Specifically, *CAMIA* (1) computes per-token prediction losses; (2) extracts signals reflecting context-dependent memorization from these losses; (3) calibrates and combines these signals into tailored membership inference tests; and (4) determines membership status based on these test outcomes. Sections 3.2 and 4 detail our signal extraction methods and their integration into *CAMIA*.

3 Context Aware Membership Signals

3.1 Intuitions

As discussed earlier, existing MIAs rely primarily on sequence-level losses, overlooking crucial token-level dynamics driving context-dependent memorization. Below, we clearly illustrate why explicitly modeling token-level prediction difficulty significantly improves MIAs for LLMs.

Traditional MIAs compare input losses to fixed thresholds, ignoring that prediction difficulty varies across data points. Easy-to-predict non-members naturally yield low losses (leading to false positives), while challenging-to-predict members often yield high losses (causing false negatives) (Ye et al., 2022; Carlini et al., 2022). Prior works address this by calibrating membership inference scores to account explicitly for input-specific difficulty (Zhang et al., 2021; Carlini et al., 2022). However, existing calibration methods typically rely on computationally expensive approaches (e.g., training the same models without the target sample) or simplistic input-level proxies (e.g., compression-based metrics (Carlini et al., 2023)), neglecting the sequential, token-level prediction behavior inherent to LLMs.

In contrast, we focus explicitly on token-level, context-dependent memorization. Specifically, LLMs predict each token sequentially based on preceding tokens (prefix contexts). Memorization emerges most strongly when the prefix provides insufficient predictive guidance, prompting models to rely heavily on memorized training data to resolve uncertainty. For example, consider predicting tokens in the sentence: *"The important thing is not to stop questioning. Curiosity has its own reason for existing."* Predicting the second occurrence of "is" is straightforward regardless of membership status,

Figure 1: Effect of token diversity calibration on membership inference (Pythia-160M, GitHub domain). Calibration clearly improves separation of members and non-members, significantly enhancing true positive rates (TPR) at low false positive rates (FPR).

Figure 2: Average token losses at the beginning (left) and end (right) of sequences (Pythia-160M, Arxiv domain). Early tokens exhibit clearer differences between members and non-members, justifying our cut-off approach.

while predicting "Curiosity"—due to its ambiguous context—is substantially easier if the model memorized this sentence during training. Such token-level distinctions highlight the importance of explicitly modeling context-dependent memorization for effective MIAs.

3.2 Signal design

Motivated by this insight, our approach calibrates membership inference directly at the token prediction level. We propose signals specifically designed to capture these subtle, context-dependent memorization patterns—previously overlooked by existing attacks (Carlini et al., 2021; Shi et al., 2023)—as detailed next.

Token diversity calibration. Texts containing repetitive patterns yield inherently lower losses regardless of memorization status (Holtzman et al.; Welleck et al.). For example, the text *"The cat sat on the mat. The cat sat on the mat."* naturally produces low loss due to repetition, potentially causing false positives (i.e., predicting a non-member as a member).

To address this bias, we introduce a lightweight calibration based on token diversity:

$$d_{\mathbf{X}} = \frac{|Dedup(\mathbf{X})|}{|\mathbf{X}|}, \quad f_{\text{Cal}}(\mathbf{X}) = \frac{\mathcal{L}(\mathbf{X}; \mathcal{M})}{d_{\mathbf{X}}}, \quad (2)$$

where $|Dedup(\mathbf{X})|$ counts unique tokens. As shown in Figure 1, this calibration better distinguishes genuinely memorized sequences from trivially predictable repetitive texts.

Token diversity calibration. Repetitive texts naturally yield low losses regardless of memorization, which can cause false positives (Holtzman et al.; Welleck et al.). For example, the sequence *"The cat sat on the mat. The cat sat on the mat."* produces low loss purely due to redundancy.

To mitigate this, we calibrate losses by token diversity:

$$d_{\mathbf{X}} = \frac{|Dedup(\mathbf{X})|}{|\mathbf{X}|}, \quad f_{\text{Cal}}(\mathbf{X}) = \frac{\mathcal{L}(\mathbf{X}; \mathcal{M})}{d_{\mathbf{X}}},$$

where $|Dedup(\mathbf{X})|$ counts unique tokens. As shown in Figure 1, this calibration improves separability between members and non-members in the GitHub domain, where repeated code patterns are common.

Filtering less informative tokens (cut-off loss). As prefixes grow longer, contextual cues reduce ambiguity and diminish memorization signals (Levy et al., 2024). For example, in Figure 2, the first few tokens predicted with little context show clear loss gaps between members and non-members, whereas later tokens converge as the prefix becomes increasingly informative.

To exploit this effect, we truncate the loss sequence to the first T' tokens and leverage the new membership signal $f_{\text{Cut}}(\mathbf{X}) = \frac{1}{T'} \sum_{t=1}^{T'} \mathcal{L}_t(x_t)$.

Loss decreasing rate (slope). When the prefix is ambiguous, memorized continuations quickly reduce uncertainty, leading to faster decreases in token losses. In other words, if the model has encountered the sequence during training, it can immediately resolve the ambiguity and drive losses down, whereas for non-members the model must rely on gradually accumulating context, resulting in a slower decline. For example, Figure 3 shows that member losses decline much more steeply than non-members.

We capture this effect by fitting a simple linear trend to the first T' token losses, where the slope serves as the signal:

$$f_{\text{Slope}}(\mathbf{X}) = \frac{\sum_{t=1}^{T'} (t - \bar{t})\big(\mathcal{L}_t(x_t) - \bar{\mathcal{L}}\big)}{\sum_{t=1}^{T'} (t - \bar{t})^2},$$

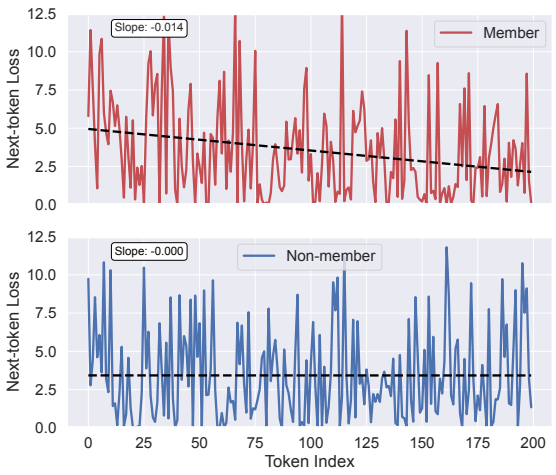

Figure 3: Linear fits to token-loss sequences (Pythia-160M, Pile-CC domain). Member losses decrease significantly faster (slope $= 0.014$) than non-members (slope $= 0.000$), reflecting stronger memorization.

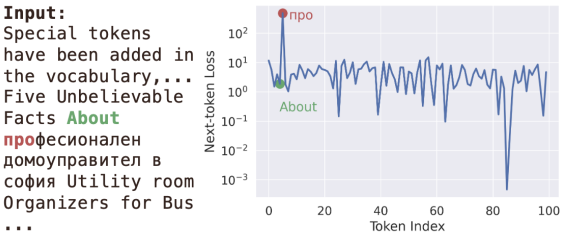

Figure 4: Example token-loss spike when the language switches from English to Bulgarian (Pythia-160M). The sudden spike (loss = 484.2) dominates the sequence, inflating the average loss to 9.33 and obscuring membership signals. This motivates using robust counts of low-loss tokens rather than relying on average loss.

with $\bar{t} = \frac{T'+1}{2}$ and $\bar{\mathcal{L}} = \frac{1}{T'}\sum_{t=1}^{T'} \mathcal{L}_t(x_t)$.

Robust low-loss counting. Average losses can be distorted by occasional spikes, for example when the input language suddenly shifts and produces extremely high token losses (Figure 4). A few "outlier" tokens result in a higher average loss value even for members, leading to a false negative error (i.e., predicting a member as a non-member).

To reduce this sensitivity, we instead count how many tokens fall below adaptive loss thresholds, thereby capturing the persistence of low-loss predictions that indicate memorization. We consider three variations: $f_{\text{CB}}(\mathbf{X}) = \frac{1}{T'}\sum_{t=1}^{T'} \mathbb{1}[\mathcal{L}_t(x_t) \leq \tau]$, which uses a fixed global threshold τ to measure the overall prevalence of low-loss tokens. $f_{\text{CBM}}(\mathbf{X}) = \frac{1}{T'}\sum_{t=1}^{T'} \mathbb{1}\left[\mathcal{L}_t(x_t) \leq \bar{\mathcal{L}}_{\mathbf{X}}\right]$, which adapts the threshold to the sequence-level mean $\bar{\mathcal{L}}_{\mathbf{X}}$, normalizing for difficulty across inputs. $f_{\text{CBPM}}(\mathbf{X}) = \frac{1}{T'}\sum_{t=1}^{T'} \mathbb{1}\left[\mathcal{L}_t(x_t) \leq \bar{\mathcal{L}}_{\mathbf{X}_{<t}}\right]$, which uses the running mean loss $\bar{\mathcal{L}}_{\mathbf{X}_{<t}}$ to capture token-level deviations relative to prior context. Together, these variations provide complementary ways of quantifying robust token-level evidence of memorization, less affected by extreme losses.

Figure 5: Distribution of loss differences after repeating inputs once (f^1_{Rep}) (Pythia-160M, Arxiv domain). Non-members benefit significantly more from repetition, exhibiting larger loss reductions than members, clearly highlighting memorization differences.

Loss fluctuation metrics. Non-members often exhibit unstable token-level predictions, reflecting unresolved uncertainty, whereas memorized sequences yield smoother and more regular loss patterns (Figure 3). Intuitively, when a model recalls training data it can make consistently confident predictions across consecutive tokens, while for unseen inputs its uncertainty fluctuates from token to token.

To quantify this distinction, we employ two sequence-complexity measures. Approximate entropy (Pincus et al., 1991) captures the degree of irregularity in local loss variations, with higher values indicating less predictable patterns typical of non-members. Lempel–Ziv complexity (Welch, 1984) measures overall compressibility of the loss sequence, where lower compressibility (i.e., higher complexity) corresponds to non-members. Both metrics thus provide complementary views of fluctuation regularity, offering robust indicators of memorization (details in Appendix A).

Amplifying signals via text repetition. Repeating an input provides extra context that the model can exploit during prediction. Intuitively, for unseen texts, the additional repetition supplies useful in-context cues, significantly reducing uncertainty, whereas for memorized texts, the model already "knows" the sequence and thus gains little benefit. For example, in Figure 5, non-members exhibit

much larger loss reductions after repetition compared to members.

We capture this difference by measuring how much the loss decreases after repeating the input once or twice: $f^1_{\text{Rep}}(\mathbf{X}) = f(\mathbf{X}; \mathcal{M}) - f(\mathbf{X}; \mathcal{M}(\mathbf{X}))$, and $f^2_{\text{Rep}}(\mathbf{X}) = f(\mathbf{X}; \mathcal{M}) - f(\mathbf{X}; \mathcal{M}([\mathbf{X}, \text{“ ”}, \mathbf{X}]))$. A larger loss reduction, therefore, provides strong evidence of non-membership.

4 MIA Test Compositions

We now compose our context-aware signals (Section 3.2) into a unified membership prediction using a hypothesis-testing framework. Our composition explicitly leverages token-level, context-dependent memorization signals to yield stronger inference.

Signal-level MIA tests. We formalize each signal as an individual hypothesis test (Sankararaman et al., 2009). The null hypothesis (H_0) assumes that the target input $\mathbf{X}$ is a non-member, so its signal values should follow the distribution observed on held-out non-member data $D_{\text{non-mem}}$. Without loss of generality, smaller signal values indicate stronger membership evidence (signals can be negated otherwise).

For each signal f, we compute an empirical p-value following the standard Monte Carlo approach (North et al., 2002; Davison and Hinkley, 1997; Long et al., 2020). Specifically, we approximate the null distribution by resampling from $D_{\text{non-mem}}$ and then evaluate the extremity of the observed statistic:

$$p_f(\mathbf{X}) = \frac{1}{|D_{\text{non-mem}}|} \sum_{\mathbf{X}' \in D_{\text{non-mem}}} \mathbb{1}\left[f(\mathbf{X}') \leq f(\mathbf{X})\right].$$

This expression is the empirical analogue of a one-sided p-value: it measures the proportion of non-member samples whose statistic is at least as extreme as that of $\mathbf{X}$. A smaller $p_f(\mathbf{X})$ therefore provides stronger evidence against H_0 and in favor of membership.

Our construction is a direct instantiation of the empirical p-value procedure widely used in permutation and bootstrap testing (North et al., 2002; Davison and Hinkley, 1997). Importantly, this goes beyond a mere CDF comparison: the statistic $f(\mathbf{X})$ is explicitly evaluated against the empirical null distribution under H_0, exactly as prescribed in standard non-parametric hypothesis testing. This framing ensures statistical validity without relying on parametric assumptions about the signal distributions, while remaining consistent with accepted practices in MIA (Long et al., 2020).

Composition of multiple tests. Given the individual signal-level p-values, we compose them into a single combined MIA test. Statistical composition methods, such as Edgington's (Edgington, 1972), Fisher's (Fisher, 1970), or George's (Mudholkar and George, 1979), aggregate the evidence across multiple signals. For instance, Edgington's method composes the p-values by summation as $p_{\text{combined}}(\mathbf{X}) = \sum_{f \in \mathcal{F}} p_f(\mathbf{X})$, predicting membership if this value is below a threshold. Our experimental results in Section 5 validate the effectiveness of this composition strategy.

Generalization with additional data. Additional labeled member data enables stronger compositions. Appendix B presents a learning-based approach using both member and non-member data, further improving attack performance.

5 Experiments

Models. We evaluate MIAs against three LLM families—*Pythia* (Biderman et al., 2023) (70M–12B parameters),*Pythia-deduped* (same sizes, trained without duplicates), and *GPT-Neo* (Black et al., 2021) (125M–2.7B parameters)—all trained on the publicly available Pile dataset (Gao et al., 2020).

Data domains and splits. We use the benchmark dataset MIMIR (Duan et al., 2024), which contains data from seven domains of the Pile dataset: Pile-CC (web), Wikipedia, PubMed Central, Arxiv, HackerNews, DM Mathematics, and GitHub. We evaluate using three MIMIR splits, each explicitly distinguishing members from non-members based on n-gram overlap.

Baseline attacks. We compare *CAMIA* with LOSS (Yeom et al., 2018), Zlib (Carlini et al., 2021), Min-K% (Shi et al., 2023), Min-K%++ (Zhang et al., 2024a), Reference-based (Carlini et al., 2021), and Neighborhood (Mattern et al., 2023) (Section 2.2). We use STABLELM-BASE-ALPHA-3B-V2 as the reference model (Duan et al., 2024), $K = 20$ for Min-K%, and 25 neighbors for Neighborhood attack.

Blind baseline. To measure potential distribution shifts arising from artificial data splits, we include a blind baseline (Das et al., 2024; Meeus et al., 2024)—a Naive Bayes classifier with bag-of-words features (Harris, 1954). This classifier, trained

Table 1: Effectiveness of attacks on the Pythia-deduped model (2.8B). We report True Positive Rate (TPR) at 1% False Positive Rate (FPR). Higher TPR indicates better attack performance. The AUC results are reported in Table 13 in Appendix C

Attack	Arxiv	Github	PubMed	HackerNews	Pile-CC	Wikipedia	Mathematics
Blind (Das et al., 2024)	0.00	32.12	0.00	1.94	2.40	0.00	**65.95**
LOSS (Yeom et al., 2018)	14.94	39.84	18.20	1.06	4.77	12.37	12.70
Zlib (Carlini et al., 2021)	10.60	46.12	14.30	2.05	5.67	9.44	8.10
Min-K% (Shi et al., 2023)	20.11	40.64	19.45	0.84	4.56	11.53	46.83
Min-K%++ (Zhang et al., 2024a)	5.20	31.91	10.44	1.43	2.87	10.24	17.94
Reference (Carlini et al., 2022)	5.86	4.68	1.22	2.63	5.93	7.36	0.00
Neighborhood (Mattern et al., 2023)	1.43	3.67	4.27	1.83	2.01	4.63	12.06
***CAMIA* (Edgington)**	23.91	**63.30**	15.78	4.86	**7.39**	10.26	26.51
***CAMIA* (George)**	**32.00**	61.33	**19.94**	**5.56**	6.76	**13.56**	20.63

solely on textual features (without any access to the target model) using an 80% train and 20% test split of the member/non-member data, and can be seen as a lower-bound baseline for MIA effectiveness in most cases.

Data access for *CAMIA*. Our primary composition approach (Section 4) uses only non-member data for calibration. Specifically, we sample an $\alpha = 30\%$ fraction of non-member test data as calibration data, with remaining non-members (70%) and an equal number of random members forming the evaluation set.

Fair comparison. For fairness, baseline attacks use the same non-member calibration set as *CAMIA*. Each baseline computes p-values from calibration data, inferring membership by thresholding these values.

Signal computation in *CAMIA*. Detailed hyperparameter settings for each signal (Section 3.2) appear in Table 11 (Appendix C). Token diversity is computed using the target model's tokenizer; results remain robust to common alternatives (e.g., OpenAI tokenizer, BPE (Sennrich, 2015), GPT-2 (Radford et al., 2019)).

Metrics. We evaluate primarily using True Positive Rate (TPR) at low False Positive Rates (FPR), capturing worst-case privacy risks more accurately than average-case metrics like AUC-ROC (Carlini et al., 2022). AUC-ROC is also reported for completeness in Appendix C.

5.1 Effectiveness of *CAMIA*

Comparison with baselines. Table 1 compares *CAMIA* with baseline attacks across seven domains using the Pythia-deduped model (2.8B). All methods are evaluated under identical conditions (i.e., datasets and calibration with non-member data only). We primarily focus on True Positive Rate (TPR) at a low False Positive Rate (1%) and also report AUC for additional context.

CAMIA consistently achieves higher TPR than baselines in almost all domains, clearly reflecting improved detection of memorized training points. For instance, on the Arxiv domain, *CAMIA* (George) achieves a TPR of 32.00%, significantly surpassing the best baseline (LOSS, 14.94%). Similarly notable improvements occur in Github (63.30% vs. 48.61%) and PubMed (19.94% vs. 19.45%).

We note one exception. Mathematics (DM), which has limited evaluation data (only 178 points), potentially causes unreliable statistical conclusions. Additionally, as explained in Section 2.3, significant distribution shift causes even the blind (model-free) baseline to perform unusually well (TPR=65.95%), clearly indicating that performance here likely reflects data distributional differences rather than model's memorization.

In HackerNews and Pile-CC, all methods (including ours) show low performance (below 8% TPR), suggesting limited memorization and inherent difficulty for MIA. We will back up our claim with experiments later (see Table 4).

Other baseline methods, such as Min-K%++ (Zhang et al., 2024a), Reference-based (Carlini et al., 2022), and Neighborhood (Mattern et al., 2023), generally have stronger assumptions or higher computational cost. For example, Min-K%++ requires full token logits, and Neighborhood performs multiple model queries per data point. Despite their higher cost, these baselines do not outperform *CAMIA* in most domains. Hence, we omit Neighborhood in subsequent experiments.

Finally, if adversaries have additional access to

Table 2: TPR at 1% FPR on the Arxiv domain (Pythia-deduped 2.8B) across different data splits. Higher substring overlap (13_gram_0.8) makes distinguishing members more difficult.

Data Split	LOSS	Zlib	Min-K%	Min-K%++	*CAMIA*
7_gram_0.2	14.94	10.60	20.11	5.20	**23.91**
13_gram_0.2	2.74	2.19	1.76	2.07	**4.00**
13_gram_0.8	0.50	0.56	0.43	**1.00**	0.41

member data, attack performance further improves, as shown in Table 13 in Appendix C.6.

Results on different models. Table 3 summarizes attack performance specifically on the Arxiv domain, covering three widely-used model families: Pythia-deduped, Pythia, and GPT-Neo. Across all model sizes and architectures, our approach *CAMIA* consistently achieves higher True Positive Rates (TPR at 1% FPR) compared to baseline attacks. This robust performance clearly demonstrates the effectiveness of explicitly capturing token-level context-dependent memorization signals. Additional results for all data domains and detailed configurations are provided in Appendix C.

Results on different data splits. Table 2 summarizes attack effectiveness (TPR at 1% FPR) on the Arxiv domain across varying substring-overlap splits from the MIMIR benchmark. As the allowed overlap increases from 7_gram_0.2 to 13_gram_0.8, distinguishing members from non-members becomes substantially more challenging, and attack performance notably decreases. On the highly challenging 13_gram_0.8 split, most methods—including *CAMIA* —perform near random guessing (1% TPR), highlighting the difficulty of membership inference under large substring overlaps. Nonetheless, *CAMIA* maintains superior or comparable performance on more distinguishable splits (7_gram_0.2, 13_gram_0.2). Detailed setups and more results are in Appendix C.

Efficiency of *CAMIA*. *CAMIA* is computationally efficient, requiring only the calculation and composition of membership signals. Evaluating 1,000 samples from the Arxiv dataset using a single A100 GPU, *CAMIA* completes in approximately 38 minutes. In comparison, the Neighborhood attack (Mattern et al., 2023) takes around 500 minutes, and the Reference-based method (Carlini et al., 2022) about 50 minutes, while simpler loss-based attacks (e.g., Zlib (Carlini et al., 2021), Min-K (Shi et al., 2023)) take roughly 25 minutes. Thus, *CAMIA* achieves superior performance at a computational cost only modestly above basic attacks, highlighting its practicality.

Table 3: TPR (1% FPR) comparison on Arxiv domain across Pythia-deduped, Pythia, and GPT-Neo models. ***CAMIA* consistently outperforms all baseline attacks.**

Family	Size	LOSS	Zlib	Min-K%	Min-K%++	Ref	*CAMIA*
Pythia (deduped)	70M	6.97	7.23	12.23	5.03	2.80	**19.54**
	1.4B	12.63	9.83	14.86	3.49	6.66	**25.23**
	2.8B	14.94	10.60	20.11	5.20	5.86	**23.91**
	6.9B	15.14	13.17	20.37	4.40	8.29	**28.69**
	12B	15.03	14.86	21.66	6.31	8.74	**28.06**
Pythia	2.8B	13.14	10.86	21.54	5.83	5.14	**24.14**
GPT-Neo	125M	9.40	6.94	9.11	3.91	2.91	**23.09**
	1.3B	12.74	11.91	15.09	4.71	5.86	**25.80**
	2.7B	16.51	14.40	21.09	7.91	7.06	**28.57**

Table 4: Impact of model size (from Pythia-deduped family) and generalization gap on *CAMIA* performance. The generalization gap is the difference between training and test losses. *CAMIA*'s performance is TPR at 1% FPR. Larger gaps correlate with increased memorization and thus better MIA performance.

Domain	Metric	160M	1.4B	2.8B	6.9B	12B
Arxiv	Gap	0.31	0.35	0.36	0.37	0.38
	TPR	23.37	25.23	25.89	28.69	28.06
Github	Gap	1.08	1.02	1.10	1.01	1.01
	TPR	41.81	54.04	60.21	55.32	61.38
HackerNews	Gap	0.09	0.10	0.11	0.11	0.12
	TPR	2.78	4.99	4.28	6.45	6.95
Pile-CC	Gap	0.07	0.10	0.10	0.13	0.15
	TPR	4.67	6.03	6.94	10.01	10.66

5.2 Ablation studies and insights

Impact of model size and generalization. Do model size and generalization ability influence *CAMIA*'s effectiveness? Table 4 compares three representative model sizes (160M, 2.8B, 12B) across various domains. We find *no direct correlation* between model size and MIA effectiveness. Instead, attack performance strongly correlates with model generalization quality, quantified by the gap between train and test losses. Specifically, domains with larger generalization gaps (e.g., GitHub; gap ≈ 1.0) reflect more significant memorization and hence, higher TPRs (up to 61.38%). Conversely, domains with smaller gaps (e.g., HackerNews; gap ≈ 0.1) exhibit limited memorization and lower TPRs (up to 6.95%). Intuitively, lower generalization ability (i.e., larger gaps between train and test losses) implies increased memorization of training

Table 5: We show the performance of our attack when using different combination methods for combining our bag of signals. The performance is measured as TPR at 1% FPR. We test on the Pythia-deduped 2.8b model.

Domain	Edgington	Fisher	Pearson	George
Arxiv	25.89	**32.11**	19.63	32.0
Mathematics	**24.92**	20.95	19.37	20.63
Github	60.21	33.03	**67.13**	61.33
PubMed	13.14	19.83	13.6	**19.94**
Hackernews	4.28	**5.67**	3.95	5.56
Pile-CC	**6.94**	6.73	6.16	6.76

data, thus amplifying privacy risks. This observation further validates our previous findings that domains such as HackerNews and Pile-CC pose low privacy risks due to low memorization.

Methods for combining signals. Table 5 compares various methods for combining p-values, including Edgington's summation (Edgington, 1972), Fisher's sum of log p-values (Fisher, 1970), Pearson's negative log of complement p-values (Pearson, 1933), and George's log-ratio method (Mudholkar and George, 1979) (Section 4). Performance slightly varies across domains, with no universally optimal method emerging—a finding consistent with prior statistical literature (Heard and Rubin-Delanchy, 2018). Edgington's simple summation approach, however, consistently achieves strong performance across domains.

Robustness to calibration set size. We evaluate *CAMIA*'s robustness across different calibration set sizes (parameterized by α). Figure 7 (Appendix C) shows stable performance for *CAMIA* across a broad range of calibration set sizes, highlighting the method's robustness even with limited calibration data.

Individual signal effectiveness. We also assess the effectiveness of individual membership signals (Section 3.2). No single signal universally performs best across all domains. For instance, token-diversity-calibrated loss (f_{Cal}) is particularly effective in specialized domains such as GitHub, repetition-amplified signals ($f^1_{\text{Rep, Cut}}$) excel in domains like Arxiv, and simple cut-off loss (f_{Cut}) performs strongly on Mathematics. This variability underscores the advantage of combining multiple signals rather than relying on any individual one. Detailed results are in Table 11 in Appendix C.

6 Additional related works

Several recent studies highlight the role of prefixes in membership leakage. He et al. (2025) examined label-only inference, showing that contextual prefixes can reveal membership but relying on a surrogate model and mainly the first token. Similarly, Meeus et al. (2025) used "canary" examples in fine-tuned models, demonstrating that high-perplexity suffixes are more easily memorized when preceded by familiar prefixes. These results support our view that ambiguous prefixes strongly drive memorization. Other works exploit conditional likelihood differences. Xie et al. (2024) proposed ReCaLL, comparing log-likelihoods conditioned on non-member prefixes. Our repetition-based amplification shares this intuition but avoids external data and integrates with other signals for robustness. Since ReCaLL did not outperform the reference attack (Carlini et al., 2021), we focus comparisons on that baseline. Beyond instance-level attacks, Maini et al. (2024) studied dataset-level inference, while Puerto et al. (2025) extended to sentence-, paragraph-, dataset-, and collection-level attacks. Our work remains at the instance-level, selectively using tokens within a sample, though our signals could serve as building blocks for broader settings. Finally, frequency-based approaches such as Zhang et al. (2024b) estimate token statistics from large reference corpora. This assumption is incompatible with our threat model, which operates with only limited non-member data, making direct comparison infeasible.

7 Conclusion

We introduce *CAMIA*, a context-aware MIA framework tailored for pre-trained LLMs. Unlike traditional MIAs, *CAMIA* captures token-level, context-dependent memorization overlooked by prior methods. Through a comprehensive evaluation, we demonstrate that *CAMIA* significantly improves attack performance compared to existing approaches. Extending *CAMIA* to evaluate fine-tuned language models and downstream applications remains a promising direction for future work.

8 Acknowledgments

We would like to thank all anonymous reviewers for their valuable comments.

9 Limitations

We acknowledge several important considerations that could influence the generalization and applicability of our findings. Below, we explicitly discuss these limitations and their potential implications.

Evaluation language limitation. Our evaluations focus on pile dataset, which contains mostly English-language data, following established benchmarks. Assessing how these findings generalize to other languages remains an important direction for future research.

Exclusion of certain models due to benchmark limitations. In this paper, we did not evaluate popular large language models such as LLaMA or GPT variants because their exact training datasets are proprietary and undisclosed. Prior works typically evaluated these models using the WikiMIA benchmark (Shi et al., 2023). However, recent critiques (Das et al., 2024; Meeus et al., 2024) demonstrate that WikiMIA introduces substantial artificial distribution shifts between members and non-members, significantly inflating MIA performance metrics. For instance, the blind attack (Das et al., 2024)—which relies solely on input data without querying the model—achieves nearly perfect performance (98.7% AUC and 94.4% TPR at 5% FPR), far surpassing the best attacks (83.9% AUC and 43.2% TPR at 5% FPR) (Zhang et al., 2024a). Due to this inherent bias, results derived from WikiMIA lack meaningful interpretability. Consequently, we chose to exclude such evaluations from our analysis. Future work should focus on developing unbiased benchmarks to rigorously evaluate membership inference attacks against models with proprietary training data.

Dependence on per-token loss information. Our membership inference attack requires access to the model's per-token loss values for a given query. While restricting access to such detailed loss information could potentially mitigate our attack, enforcing such restrictions might limit legitimate model usage scenarios and APIs that currently rely on exposing this information (e.g., for fine-tuning or debugging purposes). Understanding and balancing the trade-offs between model functionality and privacy remains an important area for future exploration.

Reliance on non-member calibration data. Our attack requires access to non-member data for calibration purposes. Although our experiments demonstrate robustness even with limited calibration data (Section 5), performance may degrade if appropriate calibration data is scarce or significantly differs from the training data distribution. Future research should further investigate methods to reduce reliance on calibration datasets.

10 Ethical Considerations

We undertake this study with a strong commitment to ethical research practices and responsible disclosure. By transparently communicating our methodology, findings, and limitations, we aim to raise awareness about privacy vulnerabilities associated with large language models. Our goal is to contribute constructively to the broader community and support ongoing efforts to balance transparency, utility, and privacy, aligning our efforts with regulatory frameworks such as the EU AI Act and U.S. AI safety policies (European Commission, 2021).

Our study exclusively utilizes publicly available datasets and models, specifically the Pythia and GPT-Neo language models (both released under the Apache 2.0 License) and the MIMIR benchmark dataset (released under the MIT License). These resources were employed strictly within their intended research purposes and license terms, and our research remains non-commercial and academic. Additionally, any artifacts created as part of this study are similarly intended solely for research purposes and are not distributed or applied beyond this scope.

We relied on comprehensive documentation provided with the Pile dataset (Gao et al., 2020)—the training corpus for Pythia and GPT-Neo—which includes detailed disclosures on domain coverage, linguistic characteristics, and acknowledged demographic and content-related biases. MIMIR, designed as a synthetic benchmark specifically for membership inference attacks, does not contain natural language content or personally identifiable information (PII). Our research did not involve the creation of new datasets with human subjects, nor did it include any form of user data collection.

To the best of our knowledge, based on the available documentation and our intended usage, the artifacts and resources employed do not include personally identifiable information (PII) or offensive content. This aligns our work with established ethical standards concerning data privacy and con-

tent safety. For data preprocessing, modeling, and evaluation tasks, we employed widely-used, open-source software packages including Hugging Face Transformers and standard Python libraries such as NumPy and SciPy. Model-specific tokenizers and default parameter configurations were used unless explicitly stated otherwise. Lastly, AI assistants (e.g., ChatGPT) were utilized to revise this manuscript. All generated content was rigorously reviewed and finalized by the authors.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE.

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

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270.

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

Anthony Christopher Davison and David Victor Hinkley. 1997. *Bootstrap methods and their application*. 1. Cambridge university press.

Paul Deutsch and Jean-Loup Gailly. 1996. zlib compressed data format specification version 3.3. In *Technical report, Network Working Group*. RFC Editor.

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

Eugene S Edgington. 1972. An additive method for combining probability values from independent experiments. *The Journal of Psychology*, 80(2):351–363.

European Commission. 2021. The EU Artificial Intelligence Act. Available at: https://artificialintelligenceact.eu/. Proposed regulation focusing on transparency and accountability in high-risk AI systems.

Ronald Aylmer Fisher. 1970. Statistical methods for research workers. In *Breakthroughs in statistics: Methodology and distribution*, pages 66–70. Springer.

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

ZS Harris. 1954. Distributional structure.

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

Nicholas A Heard and Patrick Rubin-Delanchy. 2018. Choosing between methods of combining-values. *Biometrika*, 105(1):239–246.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. *arXiv preprint arXiv:2402.14848*.

Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. 2020. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 521–534. IEEE.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024. Llm dataset inference: Did you train on my dataset? *arXiv preprint arXiv:2406.06443*.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343.

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

Matthieu Meeus, Lukas Wutschitz, Santiago Zanella-Béguelin, Shruti Tople, and Reza Shokri. 2025. The canary's echo: Auditing privacy risks of llm-generated synthetic text. *arXiv preprint arXiv:2502.14921*.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022. Quantifying privacy risks of masked language models using membership inference attacks. In *The Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Govind S Mudholkar and EO George. 1979. The logit statistic for combining probabilities-an overview. *Optimizing methods in statistics*, 345:365.

Sasi Kumar Murakonda, Reza Shokri, and George Theodorakopoulos. 2021. Quantifying the privacy risks of learning high-dimensional graphical models. In *International Conference on Artificial Intelligence and Statistics*, pages 2287–2295. PMLR.

Bernard V North, David Curtis, and Pak C Sham. 2002. A note on the calculation of empirical p values from monte carlo procedures. *The American Journal of Human Genetics*, 71(2):439–441.

Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.

Karl Pearson. 1933. On a method of determining whether a sample of size n supposed to have been drawn from a parent population having a known probability integral has probably been drawn at random. *Biometrika*, pages 379–410.

Steven M Pincus, Igor M Gladstone, and Richard A Ehrenkranz. 1991. A regularity statistic for medical data analysis. *Journal of clinical monitoring*, 7:335–345.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. 2009. Genomic privacy and limits of individual detection in a pool. *Nature genetics*, 41(9):965–967.

Rico Sennrich. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

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

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

Terry A. Welch. 1984. A technique for high-performance data compression. *Computer*, 17(06):8–19.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. 2024. ReCaLL: Membership inference via relative conditional log-likelihoods. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8671–8689, Miami, Florida, USA. Association for Computational Linguistics.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2022. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106.

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

Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. 2024. Low-cost high-power membership inference attacks. In *Forty-first International Conference on Machine Learning*.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Characterizing structural regularities of labeled data in overparameterized models. *arXiv preprint arXiv:2002.03206*.

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

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024b. Pretraining data detection for large language models: A divergence-based calibration method. In *EMNLP*.

A Measurement of fluctuations

In Section 3.2, we introduced fluctuation-based signals to quantify how the uncertainty in the token-level loss sequence varies over time, capturing patterns indicative of memorization. Here, we provide formal definitions and intuitive explanations for the two fluctuation measures: *Approximate Entropy* and *Lempel–Ziv Complexity*.

Approximate entropy (ApEn). Approximate entropy measures the unpredictability of fluctuations in the token-loss sequence by quantifying how often similar patterns recur as the sequence length increases (Pincus et al., 1991). Intuitively, ApEn checks whether short segments of the sequence remain similarly close when extended slightly longer, based on a predefined similarity tolerance r.

Formally, given a token-loss sequence $\{\mathcal{L}_t(x_t)\}_{t=1}^{T'}$, we define subsequences of length m starting at position t as:

$$u_t^m = (\mathcal{L}_t(x_t), \mathcal{L}_{t+1}(x_{t+1}), \ldots, \mathcal{L}_{t+m-1}(x_{t+m-1})).$$

We then measure the distance between two subsequences u_t^m and $u_{t'}^m$ by the maximum absolute difference among their corresponding elements:

$$d(u_t^m, u_{t'}^m) \tag{3}$$

$$= \max_{k=1,\ldots,m} \left|\mathcal{L}_{t+k-1}(x_{t+k-1}) - \mathcal{L}_{t'+k-1}(x_{t'+k-1})\right|. \tag{4}$$

Next, for each subsequence u_t^m, we calculate the proportion of subsequences within a similarity threshold r:

$$C_t^m(r) = \frac{\#\{u_{t'}^m : d(u_t^m, u_{t'}^m) \leq r\}}{T' - m + 1}$$

Then, we compute the logarithmic average across all subsequences:

$$\Phi^m(r) = \frac{1}{T' - m + 1} \sum_{t=1}^{T'-m+1} \ln C_t^m(r).$$

Finally, approximate entropy is defined as the difference between these averages at subsequence lengths m and $m+1$:

$$f_{\text{ApEn}}(\mathbf{X}) = \Phi^m(r) - \Phi^{m+1}(r).$$

In our experiments, we choose $m = 8$ and $r = 0.8$, as these parameters yielded the best performance when used individually as membership inference signals.

Lempel–Ziv complexity (LZ complexity). Lempel–Ziv complexity quantifies the diversity or complexity of patterns present in the token-loss sequence, inspired by compression-based methods (Welch, 1984). Intuitively, LZ complexity evaluates how many unique patterns exist in the sequence by breaking it down into the smallest number of non-repeating segments (phrases).

To apply Lempel–Ziv complexity to our continuous loss sequence, we first discretize the losses into bins, obtaining a sequence of bin indices $\{B_1, B_2, \ldots, B_{T'}\}$, where each B_t corresponds to the bin containing the token loss $\mathcal{L}_t(x_t)$. Formally, the Lempel–Ziv complexity is computed as:

$$f_{\text{LZ}}(\mathbf{X}) = \text{LZW}(\{B_1, B_2, \ldots, B_{T'}\}),$$

where $\text{LZW}(\cdot)$ returns the total number of unique phrases required to describe the sequence fully according to the Lempel–Ziv–Welch compression algorithm.

These fluctuation-based signals provide robust indicators of context-dependent memorization by capturing the regularity and complexity in token-level predictions.

B MIA Test Composition: Learning to Compose Signals

In Section 4, we introduced a hypothesis-testing framework for composing multiple membership inference signals, assuming the adversary has access only to non-member data. In this appendix, we extend this approach to the more powerful setting where the adversary has access to both labeled member and non-member data, referred to collectively as the *attack dataset* (D_{attack}). Below, we formalize this scenario as a supervised learning problem and describe our method in detail.

Formalization. We formulate the composition of membership signals as a supervised classification task. Given a set of membership signals $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$, we represent each target input $\mathbf{X}$ by a feature vector:

$$\mathbf{X}_{\mathcal{F}} = (f_1(\mathbf{X}), f_2(\mathbf{X}), \ldots, f_{|\mathcal{F}|}(\mathbf{X})),$$

where each element $f_i(\mathbf{X})$ is computed using the method described in Section 3.2. Our goal is to learn a model that predicts whether $\mathbf{X}$ is a member ($y = 1$) or a non-member ($y = 0$) of the target model's training data.

Specifically, we use labeled member and non-member examples in D_{attack} to train a binary classifier. The trained classifier then estimates the membership probability of any new query based on its computed feature vector.

Choice of classifier. We adopt logistic regression due to its simplicity, interpretability, and efficiency. Logistic regression learns a weight vector $\mathbf{w} \in \mathbb{R}^{|\mathcal{F}|}$ that linearly combines the signal features:

$$\hat{y} = \sigma(\langle \mathbf{X}_{\mathcal{F}}, \mathbf{w} \rangle),$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function and $\langle \cdot, \cdot \rangle$ denotes the inner product. The model parameters $\mathbf{w}$ are trained by minimizing the standard logistic loss over the attack dataset:

$$\min_{\mathbf{w}} - \sum_{(\mathbf{X},y) \in D_{\text{attack}}} \frac{y \log(\hat{y}) + (1-y)\log(1-\hat{y})}{|D_{\text{attack}}|},$$

where y is the true membership label of $\mathbf{X}$, and $\hat{y}$ is the predicted membership probability. After training, for a given target input $\mathbf{X}$, we first compute its feature vector and then apply the trained logistic regression model to predict membership status.

Dimensionality reduction with PCA. For each signal defined in Section 3.2, multiple variations can exist (e.g., varying the cut-off time for the slope signal). While including all variations could potentially enhance predictive power, it may also introduce redundancy, increasing the dimensionality unnecessarily and making the classifier less stable.

To balance predictive accuracy and complexity, we use Principal Component Analysis (PCA) (Pearson, 1901) to reduce dimensionality within each group of related signals. Specifically, for each signal group, we apply PCA to compress the set of variations into a smaller number of principal components, capturing the majority of the variability within that group. These principal components then serve as inputs to the logistic regression model, providing a more compact, effective representation for membership inference.

Overall, combining supervised learning with dimensionality reduction enables our MIA framework to leverage richer available data (both members and non-members), resulting in stronger inference performance compared to the simpler hypothesis-testing approach presented in the main text (see experimental validation in Appendix C.6).

C Additional Experiments

C.1 Extended Evaluation of *CAMIA* Across Model Families

We first provide extensive evaluations across various model families, including:

- **Pythia-deduped models (70M–12B)**: Results in Table 6.
- **GPT-Neo models (125M–2.7B)**: Results in Table 7.
- **Pythia model (2.8B)**: Results in Table 8.

Consistently, our method (*CAMIA*) achieves higher True Positive Rates (TPR) at low False Positive Rates (FPR), outperforming all baselines. Figure 6 further illustrates this superior performance via ROC curves.

C.2 Robustness to Calibration Dataset Size

Recall from Section 4 that *CAMIA* primarily uses non-member data for calibration. Figure 7 demonstrates stable attack performance (AUC) across varying calibration sizes (α). Additionally, when member data is also accessible, logistic regression (LR)-based signal combination (introduced in Appendix B) further improves performance, highlighting the benefit of richer training data.

C.3 Effectiveness Across Splits with Different Overlap Levels

Tables 9 and 10 provide results on MIMIR splits (13_gram_0.2, 13_gram_0.8) with varying membership overlap. Increased overlap reduces overall effectiveness due to ambiguity in membership. Nonetheless, *CAMIA* consistently maintains superior performance in clearly separable cases (e.g., 13_gram_0.2).

C.4 Individual Signal Performance Analysis

We present detailed results on individual membership signals from Section 3.2 in Tables 11 and 12. Specifically, we consider:

- **Cut-off loss (f_{Cut})**: Evaluated with $T' = 200, 300, T$; repetition-amplified versions $f^1_{\text{Rep,Cut}}, f^2_{\text{Rep,Cut}}$.
- **Token diversity calibrated loss (f_{Cal})**.
- **Perplexity ($f_{\text{PPL}}, f_{\text{Cal,PPL}}$)**: Standard and calibrated perplexity signals.
- **Robust counting signals ($f_{\text{CB}}, f_{\text{CBM}}, f_{\text{CBPM}}$)**.
- **Lempel–Ziv complexity (f_{LZ})**.
- **Slope of loss sequence (f_{Slope})**.
- **Approximate entropy (f_{ApEn})**.

No individual signal universally excels, emphasizing the value of combining multiple signals.

C.5 Visualizing Signal Distributions Across Domains

Figures 8 illustrates signal distributions for members and non-members across domains, providing visual confirmation that signals effectively distinguish membership, yet vary by domain.

C.6 Logistic Regression for Signal Combination (Access to Member Data)

We detail our LR-based signal combination (introduced in Appendix B). Specifically, we train a logistic regression model using both member and non-member data to combine individual signals into a unified membership prediction. For training, we sample $\alpha\%$ of both member (train set) and non-member (test set) data to form the attack dataset. The remaining data serves as the evaluation set. Results in Table 13 confirm LR-based *CAMIA* significantly enhances performance.

Impact of Dimensionality Reduction on LR. We assess dimensionality reduction (via PCA) to manage redundancy in the signal feature set. Table 14 shows that reducing each signal group's dimensions to 2 (Group PCA with $c = 2$) achieves optimal performance across most domains.

Signal Importance Analysis. Figure 9 visualizes signal importance via LR coefficients. Calibrated loss emerges as dominant in specialized domains (e.g., GitHub), while other signals contribute significantly in varied contexts (e.g., "count below previous mean" in Pile-CC), highlighting the advantage of combining diverse signals.

C.7 Relation Between Model Size, Generalization, and MIA Effectiveness

Lastly, Table 15 explores relationships between model size, generalization gap (train-test loss difference), and MIA performance. We find no direct correlation with model size; instead, larger generalization gaps strongly correlate with increased

Table 6: Effectiveness of attacks on Pythia-deduped models with different sizes. We report the AUC and the TPR (in %) at 1% FPR. The results are averaged over 10 runs across different random splits of the attack's training and test datasets. ***CAMIA* consistently outperforms prior MIAs across different domains and model sizes.**

Model Size	Attack/Baseline	Arxiv		Mathematics		Github		PubMed		HackerNews		Pile-CC	
		AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
	Blind (Das et al., 2024)	0.76	0.0	0.93	65.95	0.84	32.12	0.75	0.0	0.53	1.94	0.55	2.4
70M	LOSS (Yeom et al., 2018)	0.73	6.97	0.95	78.73	0.82	19.52	0.79	15.47	0.58	3.05	0.53	1.94
	Zlib (Carlini et al., 2021)	0.73	7.23	0.82	28.41	0.86	48.94	0.78	14.01	0.57	2.72	0.51	3.23
	Min-K% (Shi et al., 2023)	0.68	12.23	0.94	75.56	0.81	19.31	0.77	14.04	0.56	3.55	0.53	1.89
	Min-K%++ (Zhang et al., 2024a)	0.56	5.03	0.74	6.83	0.72	7.66	0.63	4.65	0.55	1.24	0.52	1.74
	Reference (Carlini et al., 2021)	0.52	2.80	0.63	2.06	0.68	4.31	0.66	6.16	0.52	0.46	0.50	2.13
	CAMIA (Edgington)	0.77	19.54	0.93	69.68	0.85	43.03	0.81	16.57	**0.59**	**4.83**	**0.53**	**4.01**
	CAMIA (LR+ Group PCA)	**0.79**	**23.37**	**0.95**	**80.95**	**0.87**	**53.46**	**0.83**	**27.85**	**0.59**	3.64	0.51	1.09
160M	LOSS (Yeom et al., 2018)	0.74	7.26	0.94	70.32	0.83	24.31	0.79	18.28	0.57	2.91	**0.54**	2.70
	Zlib (Carlini et al., 2021)	0.74	6.06	0.81	21.59	0.87	45.48	0.78	16.74	0.57	2.76	0.52	3.37
	Min-K% (Shi et al., 2023)	0.69	8.49	0.92	68.89	0.82	25.74	0.78	17.67	0.55	2.67	0.53	2.99
	Min-K%++ (Zhang et al., 2024a)	0.53	2.14	0.76	16.51	0.72	10.48	0.62	8.02	0.53	1.10	0.52	2.30
	Reference (Carlini et al., 2021)	0.57	1.09	0.62	0.16	0.68	3.51	0.68	4.04	0.51	1.04	0.52	2.64
	CAMIA (Edgington)	0.79	23.37	0.90	31.11	0.87	41.81	0.81	21.25	0.59	2.78	**0.54**	**4.67**
	CAMIA (LR+Group PCA)	**0.80**	**24.74**	**0.95**	**73.97**	**0.88**	**56.91**	**0.83**	**30.93**	**0.59**	**4.26**	0.53	1.40
1.4B	LOSS (Yeom et al., 2018)	0.77	12.63	0.92	43.49	0.86	30.05	0.78	16.16	0.59	1.99	0.55	4.56
	Zlib (Carlini et al., 2021)	0.77	9.83	0.80	15.24	0.89	36.38	0.77	13.95	0.58	2.19	0.54	5.91
	Min-K% (Shi et al., 2023)	0.74	14.86	0.93	67.14	0.85	29.95	0.78	18.05	0.57	2.14	0.55	4.70
	Min-K%++ (Zhang et al., 2024a)	0.64	3.49	0.75	15.87	0.81	22.55	0.63	8.08	0.55	1.52	0.55	3.96
	Reference (Carlini et al., 2021)	0.71	6.66	0.50	1.27	0.72	0.96	0.67	1.54	0.54	1.39	**0.59**	5.90
	CAMIA (Edgington)	**0.81**	25.23	0.83	11.90	0.89	54.04	0.79	14.22	0.60	**4.99**	0.55	**6.03**
	CAMIA (LR+ Group PCA)	**0.81**	**31.23**	**0.95**	**71.90**	**0.91**	**57.77**	**0.82**	**26.22**	**0.60**	4.55	0.55	2.74
2.8B	LOSS (Yeom et al., 2018)	0.78	14.11	0.91	19.21	0.87	39.68	0.78	18.28	*0.60*	2.03	0.55	4.63
	Zlib (Carlini et al., 2021)	0.77	10.86	0.80	11.43	0.90	42.02	0.77	14.51	0.59	2.49	0.54	5.83
	Min-K% (Shi et al., 2023)	0.75	20.63	0.92	*54.60*	0.87	40.27	0.78	20.09	0.58	1.24	0.55	4.71
	Min-K%++ (Zhang et al., 2024a)	0.65	5.71	0.72	19.84	0.84	31.49	0.66	9.80	0.57	1.52	0.54	3.27
	Reference (Carlini et al., 2021)	0.71	6.46	0.45	0.79	0.72	4.79	0.63	1.60	0.57	3.00	**0.59**	6.34
	CAMIA (Edgington)	**0.81**	25.89	0.83	24.92	0.90	60.21	0.79	13.14	**0.61**	4.28	0.55	**6.94**
	CAMIA (LR + Group PCA)	**0.81**	**32.89**	**0.95**	**72.22**	**0.91**	**64.57**	**0.82**	**26.72**	0.60	**4.46**	0.54	2.70
6.9B	LOSS (Yeom et al., 2018)	0.78	15.14	0.92	26.35	0.87	33.88	0.78	16.51	0.60	1.85	0.57	6.61
	Zlib (Carlini et al., 2021)	0.78	13.17	0.80	12.38	0.90	38.46	0.77	13.26	0.59	2.78	0.55	7.54
	Min-K% (Shi et al., 2023)	0.75	20.37	0.92	60.79	0.87	34.95	0.78	18.98	0.59	2.10	0.57	6.20
	Min-K%++ (Zhang et al., 2024a)	0.65	4.40	0.73	17.78	0.84	25.48	0.67	8.55	0.58	1.92	0.56	5.13
	Reference (Carlini et al., 2021)	0.72	8.29	0.46	2.06	0.64	0.64	0.60	1.31	0.58	1.77	**0.64**	9.87
	CAMIA (Edgington)	**0.82**	28.69	0.86	29.05	0.90	55.32	0.79	11.89	**0.61**	**6.45**	0.58	**10.01**
	CAMIA (LR + Group PCA)	**0.82**	**33.23**	**0.95**	**70.79**	**0.91**	**63.72**	**0.82**	**24.53**	**0.61**	5.12	0.57	4.31
12B	LOSS (Yeom et al., 2018)	0.79	15.03	0.92	17.30	0.88	35.05	0.77	16.54	0.61	2.14	0.58	7.14
	Zlib (Carlini et al., 2021)	0.78	14.86	0.81	9.37	0.91	36.70	0.77	11.77	0.60	3.07	0.56	8.57
	Min-K% (Shi et al., 2023)	0.77	21.66	0.92	51.11	0.88	35.21	0.78	20.99	0.60	2.43	0.58	6.49
	Min-K%++ (Zhang et al., 2024a)	0.68	6.31	0.70	22.70	0.86	27.23	0.67	9.80	0.59	1.96	0.58	6.51
	Reference (Carlini et al., 2021)	0.73	8.74	0.45	0.48	0.61	0.69	0.58	1.05	0.61	2.72	0.67	10.57
	CAMIA (Edgington)	**0.82**	28.06	0.85	27.62	0.91	61.38	0.79	11.77	**0.61**	**6.95**	**0.59**	**10.66**
	CAMIA (LR + Group PCA)	**0.82**	**36.06**	**0.95**	**69.84**	**0.92**	**63.78**	**0.82**	**21.28**	**0.61**	5.74	0.58	4.89

memorization and thus higher attack effectiveness, reinforcing generalization's importance for privacy assessment.

C.8 Impact of averaging the loss over a small window

Carlini et al. (2021) proposed computing the perplexity over a small sliding window instead of averaging the loss over all tokens. Table 16 reports the True Positive Rate (TPR) at 1% False Positive Rate (FPR) on the Pythia-2.8B model. CAMIA achieves substantially higher TPR than the sliding window approach. The latter computes perplexity only on consecutive sequences of K tokens, which does not necessarily align with where membership signals are strongest. By contrast, CAMIA explicitly identifies and leverages tokens that are most informative given their contextual uncertainty, thereby producing more effective membership inference.

Table 7: Effectiveness of attacks on GPT-Neo models with different sizes. We report the AUC and the TPR (in %) at 1% FPR. The results are averaged over 10 runs across different random splits of the attack's training and test datasets. ***CAMIA* consistently outperforms prior MIAs across different domains and model sizes.**

Model Size	Attack	Arxiv		Mathematics		Github		PubMed		HackerNews		Pile-CC	
		AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
125M	LOSS (Yeom et al., 2018)	0.76	9.40	**0.95**	**77.94**	0.83	24.84	0.81	19.53	0.57	1.59	0.53	2.29
	Zlib (Carlini et al., 2021)	0.76	6.94	0.82	27.62	0.86	39.68	0.79	21.48	0.57	2.25	0.51	2.53
	Min-K% (Shi et al., 2023)	0.72	9.11	0.94	75.24	0.82	25.43	0.80	21.13	0.56	1.52	0.53	2.61
	Min-K%++ (Zhang et al., 2024a)	0.62	3.91	0.70	16.35	0.78	18.78	0.67	10.29	0.54	2.14	0.52	2.13
	Reference (Carlini et al., 2021)	0.65	2.91	0.56	0.32	0.67	1.12	0.73	6.40	0.51	0.84	0.51	2.29
	CAMIA (Edgington)	0.81	23.09	0.88	16.51	0.87	50.53	0.82	19.94	**0.59**	2.41	**0.54**	**4.10**
	CAMIA (LR+Group PCA)	**0.82**	**28.00**	**0.95**	77.30	**0.89**	**65.85**	**0.84**	**28.90**	0.57	**3.47**	0.52	1.50
1.3B	LOSS (Yeom et al., 2018)	0.78	12.74	0.93	63.02	0.86	39.10	0.80	18.81	0.59	1.74	0.54	4.56
	Zlib (Carlini et al., 2021)	0.78	11.91	0.80	14.76	0.88	51.28	0.78	19.48	0.58	2.19	0.53	4.24
	Min-K% (Shi et al., 2023)	0.75	15.09	0.93	70.48	0.86	38.94	0.80	22.24	0.57	1.96	0.54	4.10
	Min-K%++ (Zhang et al., 2024a)	0.66	4.71	0.71	26.03	0.81	33.35	0.68	9.01	0.56	1.88	0.53	2.99
	Reference (Carlini et al., 2021)	0.71	5.86	0.49	1.43	0.66	1.97	0.70	1.31	0.52	1.66	**0.55**	4.60
	CAMIA (Edgington)	**0.82**	25.80	0.83	18.73	0.90	62.50	0.81	17.70	**0.60**	**4.17**	**0.55**	**6.16**
	CAMIA (LR+Group PCA)	**0.82**	**32.23**	**0.95**	**74.44**	**0.91**	**65.96**	**0.83**	**27.94**	0.59	3.75	0.54	2.80
2.7B	LOSS (Yeom et al., 2018)	0.79	16.51	0.93	55.71	0.87	41.86	0.80	21.25	**0.59**	1.61	0.55	4.97
	Zlib (Carlini et al., 2021)	0.78	14.40	0.81	15.87	0.89	50.32	0.78	18.63	0.58	2.43	0.54	5.34
	Min-K% (Shi et al., 2023)	0.76	21.09	0.93	69.68	0.87	42.18	0.80	23.46	0.57	1.79	0.55	4.64
	Min-K%++ (Zhang et al., 2024a)	0.66	7.91	0.72	27.14	0.83	34.20	0.69	12.18	0.57	1.96	0.54	3.79
	Reference (Carlini et al., 2021)	0.72	7.06	0.52	0.32	0.65	1.54	0.69	1.66	0.52	1.88	**0.57**	5.60
	CAMIA (Edgington)	0.82	28.57	0.86	21.75	0.91	60.59	0.81	15.84	**0.59**	2.69	0.56	**5.97**
	CAMIA (LR+Group PCA)	**0.83**	**37.03**	**0.95**	**73.65**	**0.92**	**67.50**	**0.83**	**23.63**	0.58	**4.13**	0.55	3.23

Table 8: Effectiveness of attacks on Pythia models with 2.8B. We report the AUC and the TPR (in %) at 1% FPR. The results are averaged over 10 runs across different random splits of the attack's training and test datasets.

Attack	Arxiv		Mathematics		Github		PubMed		HackerNews		Pile-CC	
	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
Loss (Yeom et al., 2018)	0.78	13.14	0.91	20.63	0.87	37.02	0.77	18.95	0.60	1.28	0.54	4.86
Zlib (Carlini et al., 2021)	0.77	10.86	0.79	11.11	0.90	42.13	0.76	15.00	0.58	2.08	0.53	6.49
MIN-K (Shi et al., 2023)	0.75	21.54	0.92	53.65	0.87	37.66	0.78	20.47	0.58	1.06	0.54	5.34
MIN-K++ (Zhang et al., 2024a)	0.65	5.83	0.70	17.46	0.84	30.85	0.66	9.71	0.57	1.46	0.54	3.29
Reference (Carlini et al., 2021)	0.71	5.14	0.44	1.27	0.73	4.89	0.62	1.28	0.57	2.65	**0.58**	**6.60**
CAMIA (Edgington)	**0.81**	24.14	0.82	27.62	**0.91**	59.63	0.79	11.95	**0.61**	3.16	0.55	6.49
CAMIA (LR + Group PCA)	**0.81**	**33.49**	**0.95**	**72.70**	**0.91**	**65.74**	**0.82**	**26.66**	0.60	**4.17**	0.54	2.81

Table 9: Effectiveness of attacks on Pythia models with size 2.8B on multiple domains for ngram_13_0.2 split. We report the AUC and the TPR at 1% FPR.

Method	Arxiv		Mathematics		Github		Hackernews		Pile CC		Pubmed Central	
	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
Blind	0.53	0.41	0.68	0.0	0.8	16.08	0.52	1.83	0.51	0.54	0.53	1.44
Loss	**0.57**	2.74	**0.68**	1.08	0.81	37.20	0.53	1.97	0.52	0.79	0.51	2.74
Zlib	0.56	2.19	0.65	2.21	0.84	48.92	0.53	1.07	0.52	1.32	0.52	2.64
MIN-K	0.56	1.76	0.65	5.66	0.81	35.85	0.53	1.30	0.53	0.69	0.52	1.97
MIN-K++	0.56	2.07	0.59	4.15	0.79	31.06	0.52	1.71	0.53	1.11	0.52	2.77
CAMIA (Edgington)	**0.57**	**4.00**	0.64	**5.88**	**0.85**	54.58	0.53	1.71	0.53	1.11	0.52	**2.81**
CAMIA (LR + Group PCA)	0.56	2.81	**0.68**	2.10	**0.85**	**59.61**	0.53	1.91	0.50	1.18	**0.53**	2.50

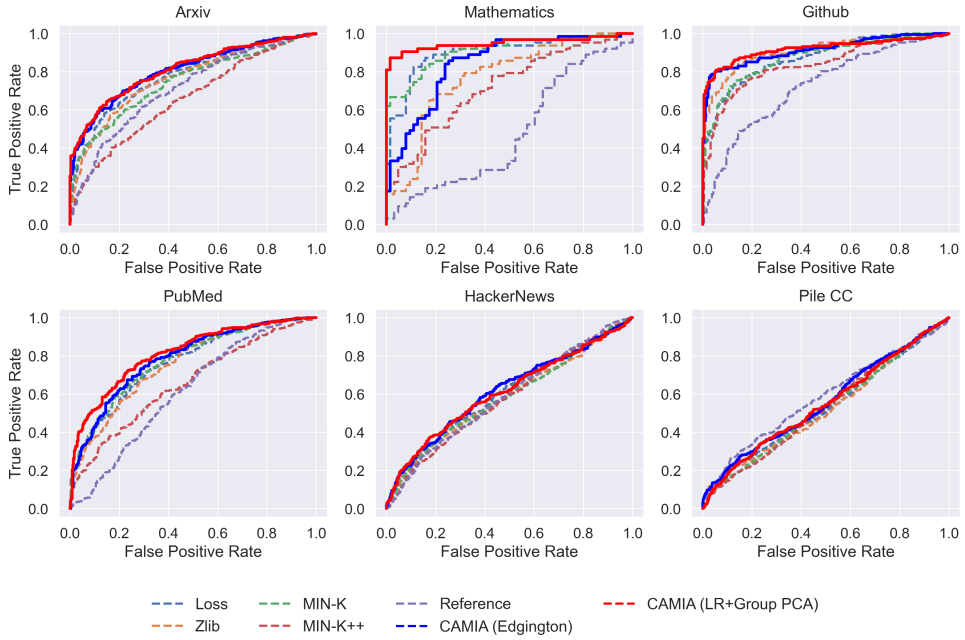

Figure 6: ROC curves comparing *CAMIA* and baseline attacks (Pythia, 2.8B).

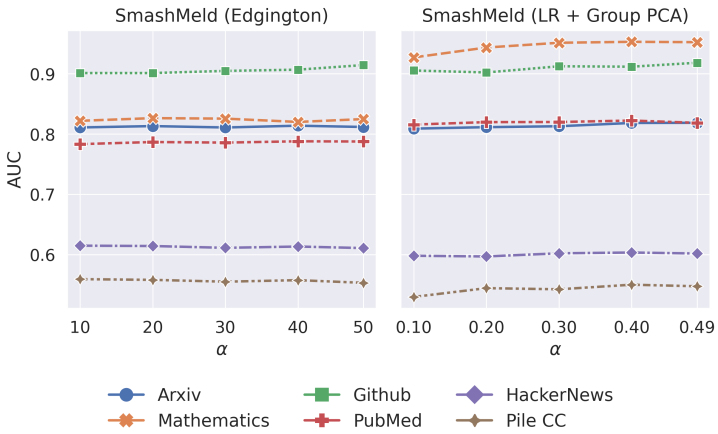

Figure 7: Effect of calibration set size (α) on *CAMIA* 's performance (Pythia-deduped, 2.8B).

Table 10: Effectiveness of attacks on Pythia-2.B models on multiple domains for ngram_13_0.8 split. We report the AUC and the TPR at 1% FPR.

	Arxiv		Mathematics		Hackernews		Pile CC		Pubmed Central	
Method	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
Bline	0.48	0.5	0.51	0.2	0.51	0.35	0.54	1.03	0.51	0.46
Loss	0.52	0.50	0.48	1.14	0.49	0.77	0.51	0.67	0.5	0.83
Zlib	0.51	0.56	0.48	1.06	0.50	0.66	0.51	1.20	0.5	1.09
MIN-K	0.52	0.43	0.49	0.44	0.50	0.74	0.52	0.73	0.5	0.83
MIN-K++	0.53	1.00	0.50	1.24	0.51	1.16	0.53	1.17	0.5	1.20
CAMIA (Edgington)	0.52	0.41	0.52	1.64	0.53	1.20	0.51	1.09	0.51	1.86
CAMIA (LR + Group PCA)	0.50	1.29	0.50	1.06	0.50	1.07	0.49	1.01	0.48	0.93

Table 11: Performance of using each individual signal. We show the TPR at 1% FPR for the Pythia-deduped model (2.8B).

Feature	Configurations	Arxiv	Mathematics	GitHub	PubMed	HackerNews	Pile-CC
f_{Cut}	$T' = T$ (Loss attack)	20.6	14.61	45.90	16.70	2.01	4.6
	$T' = 200$	32.8	**33.71**	58.58	20.98	7.89	6.5
	$T' = 300$	25.4	22.47	54.48	17.72	4.33	4.7
$f^1_{\text{Rep, Cut}}$	$T' = T$	19.0	5.62	36.57	18.74	2.17	0.8
	$T' = 200$	**40.0**	3.37	58.96	20.77	**8.82**	0.9
	$T' = 300$	24.8	3.37	51.87	18.13	4.33	0.7
$f^2_{\text{Rep, Cut}}$	$T' = T$	19.2	5.62	33.58	19.35	1.70	4.7
	$T' = 200$	33.8	3.37	57.84	20.57	**8.82**	6.4
	$T' = 300$	25.0	3.37	50.00	18.13	4.80	4.8
f_{Cal}	$T' = T$	16.2	6.74	69.78	20.37	3.56	6.5
	$T' = 200$	28.4	4.49	70.90	14.26	6.50	6.5
	$T' = 300$	20.8	4.49	72.76	13.24	2.48	**7.5**
$f^1_{\text{Rep, Cal}}$	$T' = T$	16.6	4.49	67.16	19.96	3.56	0.8
	$T' = 200$	35.2	4.49	66.79	16.70	5.26	0.9
	$T' = 300$	23.4	3.37	67.91	14.05	4.80	0.8
$f^2_{\text{Rep, Cal}}$	$T' = T$	15.8	6.74	69.40	22.00	3.56	6.9
	$T' = 200$	28.4	5.62	66.79	14.46	6.35	5.9
	$T' = 300$	21.8	3.37	68.66	13.65	2.94	7.4
f_{PPL}	$T' = T$	20.6	14.61	45.90	16.70	2.01	4.6
	$T' = 200$	32.8	33.71	58.58	20.98	7.89	6.5
	$T' = 300$	25.4	22.47	54.48	17.72	4.33	4.7
$f^1_{\text{Rep,PPL}}$	$T' = T$	20.8	2.25	36.19	16.90	1.70	0.8
	$T' = 200$	35.4	0.00	58.21	19.76	8.67	0.9
	$T' = 300$	25.8	1.12	50.37	15.07	5.42	0.5
$f^2_{\text{Rep, PPL}}$	$T' = T$	20.6	2.25	33.58	18.33	1.70	4.7
	$T' = 200$	32.6	1.12	58.21	19.76	8.51	6.4
	$T' = 300$	26.0	2.25	49.63	17.11	4.95	4.8

Table 12: Performance of using each individual signal. We show the TPR at 1% FPR for the Pythia-deduped model (2.8B).

Feature	Configurations	Arxiv	Mathematics	GitHub	PubMed	HackerNews	Pile-CC
$f_{\text{Cal, PPL}}$	$T' = T$	16.6	6.74	72.39	21.18	2.48	6.7
	$T' = 200$	34.8	4.49	**73.88**	23.22	6.04	5.9
	$T' = 300$	25.2	4.49	**73.88**	19.35	4.18	**7.5**
$f^1_{\text{Rep, Cal, PPL}}$	$T' = T$	16.6	4.49	57.46	19.14	2.79	0.8
	$T' = 200$	38.2	2.25	67.91	20.77	5.57	0.9
	$T' = 300$	27.6	2.25	65.30	15.68	4.02	0.5
$f^2_{\text{Rep, Cal, PPL}}$	$T' = T$	16.0	4.49	57.84	23.01	2.48	6.2
	$T' = 200$	36.4	2.25	67.54	21.59	5.42	6.6
	$T' = 300$	27.0	2.25	65.30	21.18	3.87	7.7
f_{CB}	$T' = 200, \tau = 1$	30.4	11.24	59.70	5.50	5.11	4.9
	$T' = 200, \tau = 2$	31.0	4.49	61.94	14.46	5.88	6.1
	$T' = 200, \tau = 3$	27.6	12.36	63.06	17.52	7.12	5.0
$f^1_{\text{Rep, CB}}$	$T' = 200, \tau = 1$	37.0	6.74	57.46	5.70	5.73	0.5
	$T' = 200, \tau = 2$	33.4	7.87	60.07	12.63	6.81	0.6
	$T' = 200, \tau = 3$	31.4	20.22	60.45	18.74	7.43	0.6
$f^2_{\text{Rep, CB}}$	$T' = 200, \tau = 1$	32.8	11.24	56.72	5.70	5.11	4.6
	$T' = 200, \tau = 2$	30.8	16.85	60.82	15.07	4.95	6.1
	$T' = 200, \tau = 3$	26.0	16.85	62.69	19.76	7.59	4.5
f_{CBM}	$T' = T$	7.6	0.00	0.00	0.81	3.25	2.4
	$T' = 200$	12.4	4.49	58.58	3.26	4.18	4.7
	$T' = 300$	14.2	2.25	39.18	3.05	3.72	2.5
$f^1_{\text{Rep, CBM}}$	$T' = T$	1.2	10.11	26.49	1.02	1.70	0.2
	$T' = 200$	18.2	5.62	51.49	2.04	1.70	0.5
	$T' = 300$	10.6	12.36	40.67	1.22	0.62	0.5
$f^2_{\text{Rep, CBM}}$	$T' = T$	1.2	8.99	14.93	0.41	1.39	1.6
	$T' = 200$	7.8	5.62	29.85	0.61	2.01	2.1
	$T' = 300$	4.2	7.87	24.25	1.02	0.62	1.7
f_{LZ}	Number of bins: 3	5.8	0.00	11.57	2.04	2.94	2.8
	Number of bins: 4	8.8	1.12	19.03	3.87	2.79	4.2
	Number of bins: 5	8.6	1.12	24.63	3.67	2.63	4.0
$f^1_{\text{Rep, LZ}}$	Number of bins: 3	6.4	0.00	39.18	1.22	1.24	1.1
	Number of bins: 4	9.4	0.00	42.16	3.05	1.08	0.7
	Number of bins: 5	9.6	1.12	42.91	4.48	2.01	0.9
$f^2_{\text{Rep, LZ}}$	Number of bins: 3	7.4	0.00	31.34	2.85	1.24	3.2
	Number of bins: 4	9.6	0.00	39.93	4.07	3.10	3.6
	Number of bins: 5	6.8	1.12	38.06	4.68	1.70	3.7
f_{CBPM}	$T' = T$	4.4	2.25	24.25	1.83	2.48	2.8
	$T' = 200$	19.2	3.37	50.75	2.65	4.64	4.4
	$T' = 300$	9.4	1.12	29.85	1.63	2.48	3.0
f_{Slope}	$T' = 600$	32.0	1.12	50.00	5.30	3.87	4.9
	$T' = 800$	27.8	13.48	42.54	**24.03**	3.41	4.6
	$T' = 1000$	20.4	3.37	50.00	21.18	2.48	3.7
f_{ApEn}	$T' = 600$	9.2	0.00	4.48	1.02	1.70	1.7
	$T' = 800$	8.8	0.00	0.37	1.63	2.01	1.8
	$T' = 1000$	10.0	0.00	0.37	1.43	3.25	2.2

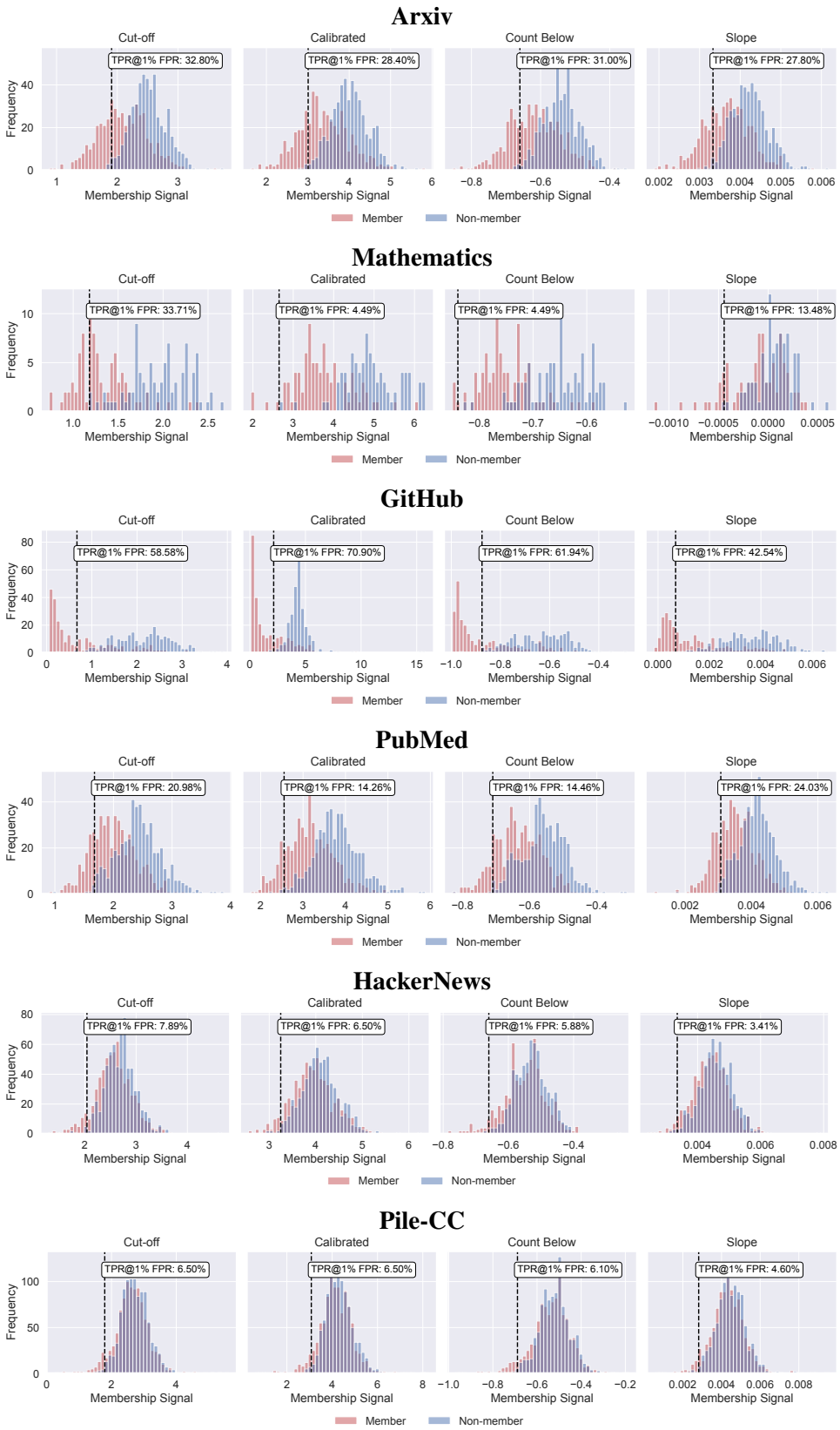

Figure 8: Membership signal distributions for easier domains (Pythia, 2.8B).

Table 13: Effectiveness of attacks on the Pythia-deduped model with 2.8B. We report the True Positive Rate (TPR) (in %) at 1% False Positive Rate (FPR) and the AUC, which quantifies the area under the TPR-FPR curve with FPR ranges from 0 to 1. Higher AUC and TPR indicate better attack performance. The results are averaged over 10 runs across different random splits of the attack's training and test datasets.

Attack	Arxiv		Mathematics		Github		PubMed		HackerNews		Pile-CC		Wikipedia	
	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
Blind (Das et al., 2024)	0.76	0.0	0.93	65.95	0.84	32.12	0.75	0.0	0.53	1.94	0.55	2.4	0.59	0.0
Use Population/Non-member Data														
LOSS (Yeom et al., 2018)	0.78	14.94	0.91	12.70	0.88	39.84	**0.79**	18.20	0.60	1.06	0.55	4.77	**0.67**	12.37
Zlib (Carlini et al., 2021)	0.78	10.60	0.82	8.10	**0.91**	46.12	0.78	14.30	0.59	2.05	0.54	5.67	0.63	9.44
Min-K% (Shi et al., 2023)	0.75	20.11	**0.93**	**46.83**	0.88	40.64	**0.79**	19.45	0.58	0.84	0.54	4.56	0.66	11.53
MIN-K%++ (Zhang et al., 2024a)	0.65	5.20	0.72	17.94	0.85	31.91	0.67	10.44	0.57	1.43	0.53	2.87	0.64	10.24
Reference (Carlini et al., 2022)	0.71	5.86	0.42	0.00	0.73	4.68	0.63	1.22	0.57	2.63	0.58	5.93	0.68	7.36
Neighborhood (Mattern et al., 2023)	0.64	1.43	0.34	12.06	0.76	3.67	0.70	4.27	0.56	1.83	0.52	2.01	0.62	4.63
CAMIA (Edgington)	**0.81**	23.91	0.84	26.51	**0.91**	**63.30**	**0.79**	15.78	**0.61**	4.86	0.55	**7.39**	0.66	10.26
CAMIA (George)	**0.81**	**32.00**	0.89	20.63	0.90	61.33	**0.79**	**19.94**	**0.61**	**5.56**	0.55	6.76	0.66	**13.56**
Use Member and Population/Non-member Data														
Loss	0.78	15.57	0.91	16.19	0.88	40.85	0.79	19.01	**0.60**	1.41	0.55	4.86	0.67	12.76
Zlib	0.78	11.37	0.82	10.63	**0.91**	47.39	0.78	14.71	0.59	2.45	0.54	5.80	0.63	9.66
MIN-K%	0.75	20.80	0.93	52.06	0.88	41.54	0.79	20.03	0.58	0.97	0.54	4.74	0.66	12.00
MIN-K%++	0.65	5.60	0.72	19.37	0.85	32.71	0.67	10.67	0.58	1.52	0.53	3.06	0.64	10.37
Reference	0.71	6.51	0.44	1.27	0.73	5.16	0.63	1.77	0.57	3.22	**0.58**	**6.30**	**0.68**	7.70
Neighborhood	0.64	1.63	0.67	51.11	0.77	5.59	0.70	3.81	0.56	1.83	0.51	2.24	0.62	4.93
CAMIA (LR)	**0.82**	**34.77**	0.93	29.68	**0.91**	**72.29**	**0.83**	**27.62**	0.59	**4.53**	0.55	4.30	0.67	**15.23**
CAMIA (LR + Group PCA)	0.81	32.89	**0.95**	**72.22**	**0.91**	64.57	0.82	26.72	**0.60**	4.46	0.54	2.70	0.66	8.96

Table 14: Effect of Dimensionality Reduction. We show the TPR (in %) at 1% FPR of *CAMIA* when the attack model (logistic regression) is trained on the raw feature, the features after pre-processed by PCA (reduce to $c = 10$ features) and group PCA (all membership signals within a group is reduced to $c = 1, 2$, and 3 features).

Domain	Raw	PCA	Group PCA		
		$c = 10$	$c = 1$	$c = 2$	$c = 3$
Arxiv	34.77	27.40	32.89	**36.94**	35.37
Mathematics	29.68	49.68	**72.22**	69.37	57.30
Github	**72.29**	62.87	64.57	66.49	67.39
PubMed	**27.62**	23.14	26.72	26.72	26.25
Hackernews	4.53	3.60	4.46	**5.47**	4.48
Pile-CC	4.30	3.73	2.70	**5.21**	4.64

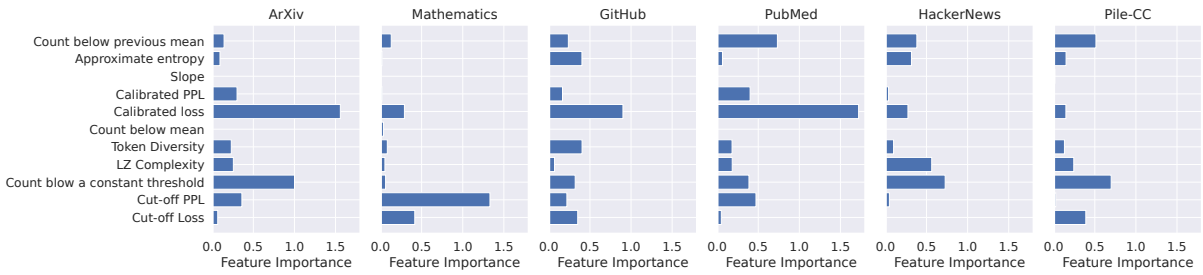

Figure 9: Logistic regression signal importance (Pythia-deduped, 2.8B).

Table 15: Model's performance and *CAMIA*'s performance of *CAMIA* on Pythia-deduped models of different sizes. The "Gap" column computes the difference between the model's losses on the training and test data For *CAMIA*, we measure its TPR (in %) at 1% FPR.

Dataset	Model	Model Performance			Performance of *CAMIA*	
		Train	Test	Gap	Edgington	LR+GPR
Arxiv	160M	2.74	3.05	0.31	23.37	24.74
	1.4B	2.12	2.47	0.35	25.23	31.23
	2.8B	1.98	2.35	0.36	25.89	32.89
	6.9B	1.92	2.29	0.37	28.69	33.23
	12B	1.86	2.24	0.38	28.06	36.06
Mathematics	160M	1.46	2.14	0.68	31.11	73.97
	1.4B	1.29	1.94	0.65	11.90	71.90
	2.8B	1.26	1.90	0.64	24.92	72.22
	6.9B	1.25	1.89	0.64	29.05	70.79
	12B	1.24	1.87	0.63	27.62	69.84
Github	160M	1.41	2.49	1.08	41.81	56.91
	1.4B	0.92	1.94	1.02	54.04	57.77
	2.8B	0.77	1.87	1.10	60.21	64.57
	6.9B	0.77	1.77	1.01	55.32	63.72
	12B	0.71	1.72	1.01	61.38	63.78
PubMed	160M	2.58	3.02	0.44	21.25	30.93
	1.4B	2.07	2.47	0.40	14.22	26.22
	2.8B	1.97	2.36	0.39	13.14	26.72
	6.9B	1.91	2.30	0.38	11.89	24.53
	12B	1.87	2.25	0.38	11.77	21.28
HackerNews	160M	3.21	3.30	0.09	2.78	4.26
	1.4B	2.60	2.70	0.10	4.99	4.55
	2.8B	2.52	2.63	0.11	4.28	4.46
	6.9B	2.40	2.51	0.11	6.45	5.12
	12B	2.33	2.45	0.12	6.95	5.74
Pile-CC	160M	3.31	3.38	0.07	4.67	1.40
	1.4B	2.66	2.76	0.10	6.03	2.74
	2.8B	2.58	2.68	0.10	6.94	2.70
	6.9B	2.43	2.56	0.13	10.01	4.31
	12B	2.36	2.51	0.15	10.66	4.89

K	Arxiv (%)	Github (%)	Pubmed (%)	Pile-CC (%)	HackerNews (%)
CAMIA	32.00	63.30	19.94	7.39	5.56
1	1.71	13.83	10.17	2.29	2.43
10	6.86	26.06	5.52	2.00	2.43
20	4.29	34.57	6.40	1.43	0.00
30	11.14	25.53	5.52	2.71	0.44
40	7.71	25.00	2.91	2.71	0.00
50	10.00	31.91	6.69	4.14	1.99
60	9.43	27.13	4.07	3.86	2.65
70	7.14	31.91	6.10	2.43	2.21
80	9.14	27.66	5.23	3.86	2.87
90	12.29	26.60	4.36	3.86	1.32
100	20.00	31.91	5.81	3.29	2.21
110	10.86	31.91	4.65	4.14	1.77
120	10.86	37.23	4.07	4.57	1.10
130	17.71	34.04	0.29	4.29	2.43
140	16.29	33.51	5.52	5.00	4.64
150	24.86	33.51	11.63	3.57	1.10
All	14.94	39.84	18.20	4.77	1.06

Table 16: TPR at 1% FPR for Pythia-2.8B under the sliding window method of Carlini et al. (2021) (window size K) compared to CAMIA.