Data Contamination Calibration for Black-box LLMs

Wentao Ye[1], Jiaqi Hu[1], Liyao Li[1], Haobo Wang[1], Gang Chen[1], Junbo Zhao[1]
[1]Zhejiang University
Correspondence: j.zhao@zju.edu.cn

Abstract

The rapid advancements of Large Language Models (LLMs) are tightly associated with the expansion of the training data size. However, the unchecked ultra-large-scale training sets introduce a series of potential risks like data contamination, i.e. the benchmark data is used for training. In this work, we propose a holistic method named **P**olarized **A**ugment **C**alibration (*PAC*) along with a brand-new dataset named StackMIA to help detect the contaminated data and diminish the contamination effect. *PAC* extends the popular MIA (Membership Inference Attack) — from the machine learning community — by forming a more global target for detecting training data to clarify invisible training data. As a pioneering work, *PAC* is very much plug-and-play that can be integrated with most (if not all) current white- and black-box (for the first time) LLMs. By extensive experiments, *PAC* outperforms existing methods by at least **4.5%**, in data contamination detection on more than **4** dataset formats, with more than **10** base LLMs. Besides, our application in real-world scenarios highlights the prominent presence of contamination and related issues. [1]

1 Introduction

As is widely acknowledged, the rapid advancements of Large Language Models (LLMs) in natural language tasks are largely attributed to the incredible expansion of the size of the training data (Kaplan et al., 2020). Despite the massive successes, this unmanaged expansion has introduced a series of significant issues that are yet explored, particularly *data contamination*. This issue arises notably when the benchmarking data is inadvertently included in the training sets. This contamination leads to misleading evaluation results (Zhou et al., 2023; Narayanan and Kapoor, 2023), thus deducing difficulties in acquiring effective and secure models. Additionally, training on datasets with copyrighted, private, or harmful content could violate laws, infringe on privacy, and introduce biases (Carlini et al., 2019; Nasr et al., 2018b). Unlike in earlier stages of machine learning, we posit that this problem would be much more prevalent in the age of the LLMs, very much due to the inevitable lack of scrutinization of the much-scaled — and often private — training data (Magar and Schwartz, 2022; Dodge et al., 2021).

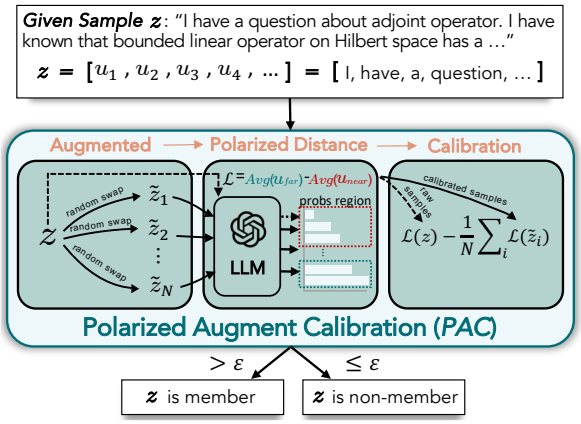

Figure 1: We focus on determining whether a given sample is contained in the training set of the LLMs. For a candidate z, *PAC* utilizes random swap augmentation to generate adjacent samples in local distribution regions. Consequently, *PAC* compares the polarized distance of z with its adjacent samples $\tilde{z}$, where the polarized distance is a spatial measurement jointly considering far and near probability regions.

In this work, we position the training data detection for LLMs as an extension of the *membership inference attacks* (MIA) (Shokri et al., 2017) in the literature of machine learning. MIA targets distinguishing whether the given data samples are *members* (training data) or *non-members* (not be trained). The previous line of work can generally be categorized into score-based (i.e., calibration-free) (Yeom et al., 2018; Salem et al., 2019; Shi et al., 2023) and calibration-based (Watson et al.,

[1]Our code is available: https://github.com/yyy01/PAC.

Findings of the Association for Computational Linguistics: ACL 2024, pages 10845–10861

2021; Carlini et al., 2022; Mattern et al., 2023) methodologies. Despite the promise, these approaches hardly suffice for current LLMs. The conventional setup in machine learning may generally focus on a small-scale training set, accompanied by global confidence distribution differences between members and non-members. However, this assumption no longer holds in LLMs, leading to a situation where non-members can also exhibit misleadingly high confidence levels, as shown in Figure 2. In addition, the MIA approaches generally rely on training an external reference or proxy model using member data distribution approximated to the targeted model. This, unfortunately, has contradicted the black-box setup of many current LLMs.

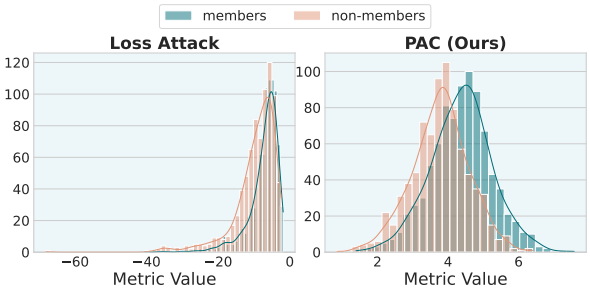

Figure 2: Histogram of the model confidence (follow the loss attack to use perplexity) before and after *PAC* in gpt-3 (davinci-002) on WikiMIA dataset (Shi et al., 2023), where *PAC* significantly enhances the salience of differences between members and non-members.

With this paper, we make two major efforts: (i)-a holistic scheme named polarized augment calibration (*PAC*) to resolve the challenges of black-box training data detection and (ii)-a brand new dataset for newly released LLMs.

Notably, the current related methods mostly assess the overfitting metric at individual points. Upon revisiting the issue of typical MIA, we find that the geometric properties of the samples may better reflect the latent differences introduced by training. For members, they should exhibit two characteristics: (i) high confidence at individual points; and (ii) being part of a poorly generalized local manifold. In other words, if a sample lies in a poor-generalization region but shows anomalous confidence, it suggests that the model is merely memorizing (or overfitting) rather than truly generalizing.

Based on such observations, as depicted in Figure 1, we propose **P**olarized **A**ugment **C**alibration (*PAC*), which explores the confidence discrepancies in local regions through data augmentation techniques, aimed at detecting those overfitted training samples. Upon this, to address the bias of existing global confidence metrics, we have developed a brand-new evaluation score named *polarized distance*, as a polarized calculation of *flagged tokens* with far and near local regions in the probability space. We further provide an in-depth explanation of *PAC* detection from a theoretical perspective. Compared to previous relevant methods relying on fully probability-accessed LLMs, we introduce a new probabilistic tracking method to extract probabilities under limited conditions(e.g., OpenAI API) for the first time.2 This becomes the trailblazers to extend our detection to a black-box setup where only access to partial probabilities is permitted, enabling *PAC* to be applied to almost all LLMs.

For the benchmark, the current available benchmark for the contamination problem is very limited. We construct and introduce StackMIA tailored for reliable and scalable detection of the latest LLMs. It ensures reliability by adopting a time-based member/non-member classification similar to (Shi et al., 2023), where members are visible during pre-training and non-members are not. StackMIA is dynamically updated, by offering detailed timestamps in order to quickly adapt to any up-to-date LLM through our curation pipeline. By contrast, the existing benchmark WikiMIA(Shi et al., 2023) does not possess these properties, making it hard to use, especially on the LLMs released post-2023.

Last but not least, we exhibit extensive experiments on 10 commonly used models to evaluate the *PAC* against six existing major baselines. These results demonstrate that *PAC* outperforms the strongest existing baseline by **5.9%** and **4.5%** in the AUC score respectively on StackMIA and WikiMIA. *PAC* further expresses superior robustness under conditions of ambiguous memory or detection of fine-tuning data. To further validate the effectiveness of *PAC* in real-world applications, we provide a set of case studies for data contamination and different security risks (Carlini et al., 2019; Nasr et al., 2018b,a) on ChatGPT and GPT-4. The conclusions from these examples highlight the ubiquity of security risks upon widespread deployment (Ye et al., 2023).

Our contributions can be summarized as:

1. We introduce *PAC*, an innovative, theory-supported MIA method for black-box LLMs that does not rely on external models, consis-

tently outperforming leading approaches.

2. Our black-box probability extraction algorithm makes *PAC* a trailblazer for LLMs with restricted probability access. We also unveil the StackMIA benchmark, addressing the gaps in MIA datasets for pretraining phrases.

3. Applying *PAC* to ChatGPT and GPT-4 highlights the prevalent data contamination issue, prompting a call to the academic community for solutions to ensure safer, more dependable LLMs.

2 Related Work

Data Contamination. As Magar and Schwartz (2022) mentioned, data contamination is the infiltration of downstream test data into the pretraining corpus, which may seriously mislead evaluation results. Dodge et al. (2021) and Brown et al. (2020) highlighted the tangible presence of contamination issues in models (e.g. GPT-3) and corpora (e.g. C4). Such contamination risks models memorizing rather than learning, affecting their exploitation (Magar and Schwartz, 2022). Detection methods have been proposed, using n-gram overlap ratios against pre-training data to identify contaminated samples (Du et al., 2022; Wei et al., 2021; Chowdhery et al., 2023). However, these existing methods rely on access to the pretraining corpus, which is unfeasible for many LLMs. Recent studies (Golchin and Surdeanu, 2023; Weller et al., 2023) shifted towards a more common black-box setting, by extracting outputs with trigger tokens (Carlini et al., 2021) or specified prompts (Sainz et al., 2023; Nasr et al.) from LLMs as contaminated samples and comparing with the test set. Unfortunately, the extracted samples are usually too broad, making these methods ineffective for detecting given samples.

Membership Inference Attack. Membership Inference Attacks (MIA), proposed by Shokri et al. (2017), is defined as determining whether a given sample is part of the training set. MIA is predicated on the models' inevitable overfitting (Yeom et al., 2018), leading to a differential performance on training samples (members) versus non-members. MIA has been applied in privacy protection (Jayaraman and Evans, 2019; Zanella-Béguelin et al., 2020; Nasr et al., 2021, 2023; Steinke et al., 2023), machine-generated text detection (Mitchell et al., 2023; Solaiman et al., 2019), DNA inference (Zerhouni and Nabel, 2008), etc. Given the limited access to most current LLMs, research has concentrated on black-box conditions. Typical score-based methods employ loss or partial token probabilities (Shi et al., 2023), though these may incorrectly flag non-members (Carlini et al., 2022). Watson et al. (2021) proposed calibration-based methods to utilize a difficulty calibration score to regularize raw scores. Specifically, Carlini et al. (2021), Ye et al. (2022), and Mireshghallah et al. (2022) train reference models to rectify anomalies with the average of different models. Mattern et al. (2023) might be most closely aligned with our work, utilizing additional models to generate similar samples for calibration purposes. while Yet, the practicality is constrained by their need for extra models. Furthermore, obtaining the necessary probabilities for MIA is challenging in many recent LLMs. To our knowledge, we are the pioneers in developing a calibration-based detection method jointly considering the calibration of global confidence and local distribution.

3 Problem Definition and Preliminary Work

We first provide a comprehensive formal definition of the training data detection for LLMs in the context of a two-stage training process (Kenton and Toutanova, 2019). Upon this foundation, we introduce a new dynamic data benchmark (Section 3.2) that can be applied to recently released LLMs. By open-sourcing the benchmark upon publication, we expect to use it to foster further research in the community.

3.1 Problem Formulation

Two-stage training. The training process of LLMs consists of two stages: unsupervised pre-training on a large-scale corpora, and supervised fine-tuning of labeled data for downstream tasks.

Given training set $\mathcal{D} = \{z_i\}_{i \in |\mathcal{D}|}$, z_i for the fine-tuning stage can be further represented as $x_i \cup y_i$ (x, y respectively represent input and label). Both z_i and x_i can be denoted as a tokens sequence $\{u_j\}_{j \in |x_i| or |z_i|}$. Generally following (Brown et al., 2020), the LM f_θ is trained in two stages by maximizing the following likelihood separately:

$$\begin{aligned} L_{pt}(\mathcal{D}) &= \sum_i \sum_j \log f_\theta(u_j | u_1, \cdots u_{j-1}) \\ L_{ft}(\mathcal{D}) &= \sum_i \log f_\theta(y_i | u_1, \cdots, u_{|x_i|}) \end{aligned} \quad (1)$$

where i, j respectively denote the index of the sample z and the index of the token within z.

Training Data Detection. Following the settings of MIA (Yeom et al., 2018; Mattern et al., 2023), for a given the target model f_θ and the sample z, the objective of training data detection can be defined as learning a detector $\mathcal{A} : (z, f_\theta) \rightarrow \{0, 1\}$, where 1 denotes that z is member data ($z \in \mathcal{D}$) and 0 denotes that $z \notin \mathcal{D}$. As mentioned in Section 2, widely used calibration-free detection methods directly construct computational scores (denoted as $\mathcal{L}$), such as the loss, and threshold them:

$$\mathcal{A}(z, f_\theta) = \mathbb{1}(\mathcal{L}(z, f_\theta) > \epsilon) \quad (2)$$

By comparing $\mathcal{L}$ with a predefined threshold ϵ, $\mathcal{A}$ can achieve the detection of members or non-members. For more recent calibration-based methods, a calibration function c is additionally introduced to correct for the detector's bias relative to the target distribution. Calibration is typically achieved by correcting the original scores with the calibration function. The calibration function is usually a designed score $c : (\tilde{z}, f_\phi) \rightarrow \mathbb{R}$ based on the differential performance on adjacent samples $\tilde{z}$ (Mattern et al., 2023) or reference models f_ϕ (Watson et al., 2021). The reference models are usually additional models trained on a similar training data distribution. Consequently, the detector $\mathcal{A}$ can be extended as:

$$\mathcal{A}(z, f_\theta) = \mathbb{1}(\mathcal{L}(z, f_\theta) - c(\tilde{z}, f_\phi) > \epsilon) \quad (3)$$

Following the standard setup (Shi et al., 2023), we primarily constrain detecting pre-training samples under black-box settings. Specifically, the detection during the fine-tuning stage will be further discussed in Section 5.6.

3.2 Dynamic Benchmark Construction

StackMIA[2] is based on the Stack Exchange dataset [3], which is widely used for pre-training. Specifically, we organize member and non-member data with fine-grained release times to ensure reliability and applicability to newly released LLMs. More Details are provided in Appendix A.

Data collection and organization. We utilize the data source provided by the official (Appendix A). Each record contains a post and answers from different users. **Data collection:** Following the data selection strategy mentioned by LLMs such as LLaMA (Touvron et al.), etc, we retain data from the 20 largest websites, including common themes like English, math, etc. Subsequently, based on the training timelines of most LLMs (Zhao et al., 2023), we set January 1, 2017, as the latest cutoff date for member data, i.e., data with both posts and answers dated before this time are considered members. For non-members, January 1, 2023, is set as the earliest occurrence time. Lastly, we build an automatic pipeline to remove HTML tags from the text. **Data filtering:** Considering the potential differences in answer sorting strategies (Ouyang et al., 2022), we only retain the post records. We apply automatic filtering based on: (1) posts contain only texts, not formulas or code; (2) posts have not been asked repeatedly. **Data organization:** To ensure applicability to a vast array of LLMs released after 2023, we reorganize the non-member data with fine-grained precision. We selected approximately 2000 posts created within each month, using the month as a cutoff point. This allows future users to construct subsets of StackMIA suitable for newly released LLMs through our provided pipeline.

Benchmark test. Following the settings of (Shi et al., 2023), we divide the original WikiMIA by length. Simultaneously, we construct **StackMIAsub**[4] dataset (8267 samples in total, Appendix A) to conduct experiments in this paper. Specifically, we set May 1, 2023, as the cutoff date for non-member data. Following (Shi et al., 2023), we sequentially select approximately balanced sets of member and non-member data by length. Additionally, we use GPT-3.5-turbo[5] to construct (Appendix A) synonymous rewritten data to test the stability of the methods under the condition of approximate memory (Ishihara, 2023). Referring to (Ippolito et al., 2022), we set BLEU (Papineni et al., 2002)> 0.75 as a condition to ensure semantic consistency in the rewritten data.

4 Methodology

We introduce **P**olarized **A**ugment **C**alibration (***PAC***), an efficient and novel calibration-based training data detection method. The key idea is to construct adjacent samples using easy data augmentation for calibrating a generalized distribution and design a brand-new polarized distance to enhance the salience. We further propose a probabilistic tracking method suitable for models with partially

[2]The StackMIAsub benchmark dataset is available here: https://huggingface.co/datasets/darklight03/StackMIA

[3]https://archive.org/details/stackexchange

[4]The StackMIAsub benchmark dataset is available here: https://huggingface.co/datasets/darklight03/StackMIAsub

[5]https://platform.openai.com/docs/api-reference

inaccessible logits for the first time.

4.1 Generating Adjacent Samples

As mentioned in Section 3.1, for a given data point z, we construct an adjacent sample space $\tilde{z}$ for the calibration function c. We opt for a simpler word-level perturbation approach through the Easy Data Augment (Wei and Zou, 2019) framework to generate $\tilde{z}$, which are adjacent in the local distribution with z. The calibration through these adjacent data points prevents calibration-free scores from being confused with misleading high-confidence, especially when the model provides a well-generalized distribution of non-members (Choquette-Choo et al., 2021). Specifically, we randomly swap 2 tokens from z, and repeat this process m times:

$$\tilde{z} = \sigma_m(z) = \sigma_m(\{u_j\}_{j\in[1,|z|]}) \tag{4}$$

where σ_m represents a bijection from z to itself (i.e., a permutation) after m random swaps. Further, different from previous works, the augmentation-based scheme focuses the calibration on local distribution and is much more efficient due to the avoidance of introducing additional models.

4.2 Polarized Distance

Due to the challenges in constructing reference models for LLMs, we calibrate directly using the difference between $\tilde{z}$ and z without introducing costly external models. As mentioned in Section 4.1, augmentation-based $\tilde{z}$ tends to exhibit non-member characteristics more. In practice, using traditional confidence scores, e.g. loss or perplexity, as the $\mathcal{L}$ score fails to demonstrate stable significance. (Shi et al., 2023) proposes to improve classification effectiveness by calculating only a portion of the low-probability outlier words. We integrate this technique and expand it to focus simultaneously on far and close local regions of the token probability, achieving a significant measurement in probability space. Specifically, consider a sequence of tokens for a given sample z, denoted as $z = \{u_j\}_{j\in[1,|z|]}$. According to Equation 1, the log-probability of each token u_i can be denoted as $\log f_\theta(u_i|u_1, \cdots, u_{i-1})$. As depicted in Figure 1, we then sort the probabilities of each token and select the largest $k_1\%$ and the smallest $k_2\%$ to form sets, denoted as $\mathrm{MAX}(z, k_1)$ and $\mathrm{MIN}(z, k_2)$, respectively. Subsequently, the polarized distance $\mathcal{L}_M$ can be denoted as:

$$\begin{aligned}\mathcal{L}_M = &\frac{1}{K_1}\sum_{u_i\in\mathrm{MAX}(z,k_1)} \log f_\theta(u_i|u_1,\cdots,u_{i-1}) \\ &-\frac{1}{K_2}\sum_{u_i\in\mathrm{MIN}(z,k_2)} \log f_\theta(u_i|u_1,\cdots,u_{i-1})\end{aligned} \tag{5}$$

where K_1 and K_2 denotes the size of $\mathrm{MAX}(z, k_1)$ and $\mathrm{MIN}(z, k_2)$ set separately.

General. According to the previous sections, the implementation of *PAC* can be represented as:

$$\mathcal{A}(z, f_\theta) = \mathbb{1}[\mathcal{L}_M(z, f_\theta) - \sum^{N} \frac{\mathcal{L}_M(\sigma_m(z), f_\phi)}{N} > \epsilon] \tag{6}$$

where N denotes the number of repetitions to reduce random errors.

4.3 Theoretical Analysis

The explanation for the approach of *PAC* is quite straightforward: through carefully designed discrete perturbations, it makes $\tilde{z}$ (i.e., $\sigma_m(z)$) exhibit non-member characteristics. Since LLMs typically rely on original natural corpora, such perturbations can confuse the model (Jin et al., 2020; Morris et al., 2020; Li et al., 2021), thereby obtaining measurable differences between z and $\tilde{z}$. By calculating the difference in the probability distribution of far and near local regions (expressed as the highest and lowest probabilities), $\mathcal{L}_M$ can reflect the model's predictive uncertainty (Duan et al., 2023) and volatility. Given a sample z, when it occurs:

$$\mathcal{L}_M(z, f_\theta) \gg \mathcal{L}_M(\sigma_m(z), f_\theta) \tag{7}$$

This means that the impact of the perturbations is significant, which indicates that z may be overfitting. In this case, z will be classified as a member sample. Furthermore, we provide a simple detailed mathematical proof in Appendix C.

4.4 Black-box Probabilistic Tracking

Almost all detection methods require accessing the probabilities of all tokens for z, which is not feasible for some current black-box models, such as GPT-4 (Achiam et al., 2023) accessed by official API. These models only provide a log-probability query interface for the top n words, where $n \leqslant 5$ usually. To address this issue, we take the GPT models as an example and construct a black-box probabilistic tracking algorithm using the logit_bias[6] function provided by the OpenAI

[6]https://platform.openai.com/docs/api-reference/completions/create#completions-create-logit_bias

API to track probability outputs. Such a function allows for setting biases of the logits for specific token IDs, which can be obtained through the tiktoken library[7]. Utilizing this feature, for each token in turn $u_i \in z$, we enumerate biases added to the corresponding token ID until the top n probability query results are just altered. The obtained bias threshold γ_i can be approximated as the difference between the log-probability of u_i and the known token u_τ (see Appendix D for the proof). Thus, the probabilities of all tokens in z can be obtained by:

$$\log f_\theta(u_i|\cdot) = \log f_\theta(u_\tau|\cdot) - \gamma_i \tag{8}$$

where $\cdot$ represents the prefix tokens of u_i. Due to the monotonicity of the impact of bias growth on query results, the enumeration process can be optimized for the binary search. Thus, we achieve a reduction in time complexity from $O(N)$ to $O(\log N)$ for obtaining the log probability of a single token with a logit of N, which is cost-effective and efficient.

With this extraction method, *PAC* becomes the first method capable of detecting training data from almost any black-box LLMs.

4.5 Pseudo Code

We provide a simple pseudocode (as shown in Algorithm 1) to illustrate the specific implementation steps of *PAC*.

Algorithm 1 Polarized Augment Calibration

Input: given data sample $z = \{u_i\}_{i\in|z|}$, source model f_θ, and decision threshold ϵ
Output: True – z is a member sample, False – z is a non-member sample.

Get the augmented sample $\tilde{z}$ with random swap, repeating m times
Select the highest K_1 probability tokens and lowerst K_2 probability tokens to construct MAX and MIN set
Calculate polarized distance $\mathcal{L}_M(z)$
$\mathcal{L} \leftarrow \frac{1}{K_1} \sum_{\text{MAX}} \log f_\theta(u|\cdot) - \frac{1}{K_2} \sum_{\text{MIN}} \log f_\theta(u|\cdot)$
$\mathbf{d} \leftarrow \mathcal{L}_M(z) - \mathcal{L}_M(\tilde{z})$
return True if $\mathbf{d} > \epsilon$ else False

[7]https://github.com/openai/tiktoken

5 Experiments

5.1 Experiments Settings

Baseline Methods: We selected six popular methods to evaluate our approach: four calibration-based and two calibration-free. Calibration-based methods include: the Neighborhood attack (**Neighbor**) (Mattern et al., 2023), which assesses loss differences between original samples and their neighbors generated by masked language models; and perplexity-based calibration (Carlini et al., 2021) techniques utilizing Zlib entropy (**Zlib**) (Gailly and Adler, 2004), lowercased sample perplexity (**Lower**), and comparisons with reference models trained on the same dataset (**Ref**). Calibration-free methods comprise the **Min-K%** method (Shi et al., 2023), predicting pre-trained samples through low-probability outlier words; and the Loss Attack (Yeom et al., 2018), substituting loss with Perplexity (**PPL**) in LLMs.

Datasets and Metric. We utilize the **StackMIA-Asub** benchmark (Section 3.2) and the **WikiMIA** dataset proposed by (Shi et al., 2023). WikiMIA (Appendix B) leverages Wikipedia timestamps and model release dates to identify member and non-member data sets, applicable for LLMs trained up to 2023. Both datasets are transformed into two formats as the guidelines in Section 3.2: the original format (**ori**) and the synonym rewritten format (**syn**).

For evaluation, we follow (Mattern et al., 2023; Carlini et al., 2022; Watson et al., 2021) and plot the ROC curve analysis method. To facilitate numerical comparison, we primarily use the **AUC** score (Area Under the ROC Curve). The AUC score (Appendix F), independent of any specific threshold, accurately gauges the method's ability to differentiate between members and non-members. It also eliminates bias from threshold selection.

Models. We conduct experiments against 10 commonly used LLMs. Six models are applicable for both WikiMIA and StackMIA, including LLaMA-13B (Touvron et al.), LLaMA2-13B (Touvron et al., 2023), GPT-J-6B (Wang and Komatsuzaki, 2021), GPT-Neo-2.7B (Black et al., 2021), OPT-6.7B (Zhang et al., 2022), and Pythia-6.9B (Biderman et al., 2023). The two GPT-3 base models, Davinci-002 and Baggage-002 (Ouyang et al., 2022), are suited for the WikiMIA dataset. Additionally, two newer models are applicable for StackMIA, including StableLM-7B (Tow et al., 2023) and Falcon-7B (Almazrouei et al., 2023).

5.2 Implements

According to Section 4, the key hyper-parameters affecting the *PAC* include the times of perturbations m, the tokens ratio in min-max distance k_1 and k_2, and the number of adjacent samples N. To ensure efficiency, N is globally fixed at 5. Based on this, we conduct a grid search (Liashchynskyi and Liashchynskyi, 2019) on a reserved small-scale validation set of StackMIA using LLaMA-13B. The final settings are $k_1 = 5$, $k_2 = 30$, and $m = 0.3 \times |z|$, where $|z|$ denotes the token number of z.

5.3 *PAC* as A More Effective Detector

Based on the settings described in Section 5.1, the primary comparison results between *PAC* and the baseline methods are listed in Table 1. The experimental outcomes indicate that *PAC* consistently outperforms across all models and all data formats. Specifically, *PAC* shows an average AUC score improvement of 4.5% on WikiMIA and 5.9% on StackMIAsub compared to all other baseline methods. Moreover, *PAC* maintains robust performance even under the conditions of synonymously approximate memories. Notably, the Min-K% method exhibits the second-best performance in all settings, validating the reliability of using local regions of token probabilities. And different from previous methods, *PAC* exhibits prominence in member recognition (Figure 2). In summary, *PAC* is an effective and versatile solution for detecting pre-training data of LLMs.

5.4 Ablation Study

To further validate the design of *augment calibration* and *polarized distance*, we conduct ablation studies on (1) methods for generating adjacent samples, and (2) metric scores. We limit the model to LLaMA-13B as an example.

Generation method. We evaluate the performance of four popular methods to generate adjacent samples. Table 2 demonstrates a clear conclusion: augmentation based on random swaps is far more effective than any other generation method. We believe the underlying reason is that the swap operation ensures better non-member attributes, making the metric more significant.

Metric Score. Similarly, we compared different metric scores in terms of their significance in difficulty calibration. As shown in Table 2, the *polarized distance* more readily facilitates the distinction between members and non-members.

5.5 Analysis Study

To explore the factors influencing the detection, we focus on the four aspects, using LLaMA-13B and Pythia series as examples. All results are shown in Figure 3.

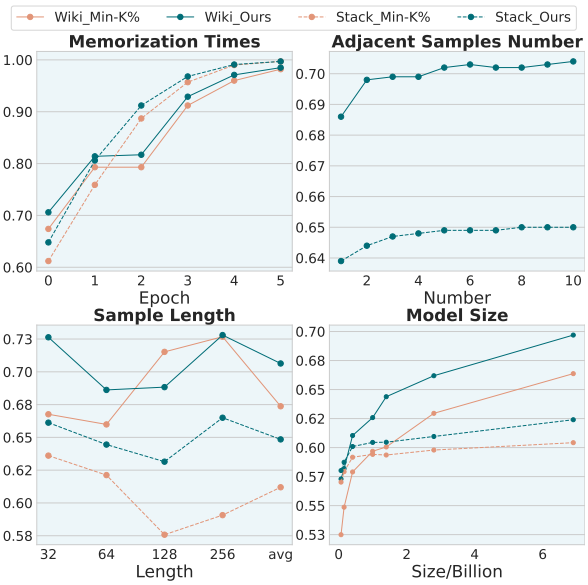

Figure 3: The AUC results as four different factors vary.

Memorization times. We employ continual pre-training to increase the times that a member is seen by the model. The results show that detection difficulty decreases with memorization times increase, likely due to an increase in the degree of overfitting.

Number of adjacent samples. We varied the number of adjacent samples from one to twice the number of adjacent samples we used. The results demonstrate that with an increase in quantity, *PAC* maintain robust performance after initial improvements.

Sample Length. As the length of samples varies, *PAC* tends to achieve better performance in both lower and higher lengths. This is likely because they respectively contain more distinctive features and more information, making detection easier.

Model Size. The performance of *PAC* continuously improves with an increase in model parameters. This may be due to larger models having a stronger learning capacity within constant training iterations.

5.6 *PAC* as A Two-Stage Detector

As more developers utilize various domain-specific data to fine-tune the same foundational models, the expansion of detection capabilities to the fine-tuning stage becomes increasingly critical. Limited

Model	Form	WikiMIA							StackMIAsub						
		PPL	Zlib	Lower	Ref	Neighbor	Min-K%	**Ours**	PPL	Zlib	Lower	Ref	Neighbor	Min-K%	**Ours**
LLaMA	ori	0.664	0.632	0.563	0.604	0.617	0.674	**0.706**$^{\uparrow 4.7\%}$	0.605	0.556	0.532	0.487	0.552	0.612	**0.648**$^{\uparrow 5.9\%}$
	syn	0.684	0.627	0.545	0.587	0.610	0.681	**0.728**$^{\uparrow 6.4\%}$	0.564	0.534	0.517	0.492	0.528	0.565	**0.592**$^{\uparrow 4.7\%}$
LLaMA2	ori	0.540	0.555	0.520	0.540	0.509	0.535	**0.560**$^{\uparrow 0.9\%}$	0.602	0.555	0.529	0.499	0.549	0.610	**0.643**$^{\uparrow 5.4\%}$
	syn	0.556	0.558	0.508	0.528	0.506	0.546	**0.572**$^{\uparrow 2.5\%}$	0.563	0.533	0.519	0.507	0.525	0.565	**0.592**$^{\uparrow 4.7\%}$
GPT-J	ori	0.641	0.620	0.558	0.616	0.631	0.675	**0.681**$^{\uparrow 0.9\%}$	0.584	0.549	0.526	0.537	0.550	0.595	**0.605**$^{\uparrow 1.7\%}$
	syn	0.632	0.602	0.545	0.599	0.605	0.644	**0.666**$^{\uparrow 3.4\%}$	0.544	0.525	0.507	0.548	0.518	0.549	**0.560**$^{\uparrow 2.0\%}$
GPT-Neo	ori	0.616	0.603	0.560	0.593	0.619	0.648	**0.666**$^{\uparrow 2.8\%}$	0.579	0.547	0.531	0.538	0.555	0.590	**0.600**$^{\uparrow 1.7\%}$
	syn	0.610	0.590	0.561	0.579	0.596	0.631	**0.653**$^{\uparrow 3.5\%}$	0.539	0.523	0.507	0.545	0.520	0.545	**0.556**$^{\uparrow 2.0\%}$
OPT	ori	0.602	0.591	0.560	0.633	0.577	0.625	**0.648**$^{\uparrow 3.7\%}$	0.602	0.558	0.533	0.492	0.583	0.607	**0.619**$^{\uparrow 2.0\%}$
	syn	0.603	0.584	0.551	0.643	0.577	0.619	**0.646**$^{\uparrow 4.4\%}$	0.559	0.534	0.518	0.508	0.545	0.560	**0.572**$^{\uparrow 2.1\%}$
Pythia	ori	0.635	0.617	0.550	0.629	0.626	0.664	**0.697**$^{\uparrow 5.0\%}$	0.598	0.557	0.532	0.549	0.559	0.604	**0.624**$^{\uparrow 3.3\%}$
	syn	0.634	0.602	0.549	0.623	0.614	0.642	**0.696**$^{\uparrow 8.4\%}$	0.553	0.532	0.515	0.559	0.525	0.556	**0.578**$^{\uparrow 4.0\%}$
StableLM	ori	-	-	-	-	-	-	-	0.515	0.506	0.449	0.482	0.518	0.510	**0.589**$^{\uparrow 14\%}$
	syn	-	-	-	-	-	-	-	0.491	0.488	0.437	0.484	0.501	0.487	**0.576**$^{\uparrow 15\%}$
Falcon	ori	-	-	-	-	-	-	-	0.613	0.566	0.519	0.577	0.573	0.617	**0.641**$^{\uparrow 3.9\%}$
	syn	-	-	-	-	-	-	-	0.569	0.541	0.505	0.588	0.537	0.569	**0.593**$^{\uparrow 0.8\%}$
davinci	ori	0.638	0.621	0.497	0.554	0.607	0.656	**0.694**$^{\uparrow 5.8\%}$	-	-	-	-	-	-	-
	syn	0.654	0.616	0.507	0.564	0.608	0.651	**0.691**$^{\uparrow 5.6\%}$	-	-	-	-	-	-	-
babbage	ori	0.569	0.575	0.492	0.475	0.537	0.559	**0.607**$^{\uparrow 6.7\%}$	-	-	-	-	-	-	-
	syn	0.582	0.576	0.513	0.483	0.540	0.574	**0.621**$^{\uparrow 6.7\%}$	-	-	-	-	-	-	-
Mean	ori	0.613	0.602	0.537	0.581	0.590	0.629	**0.657**$^{\uparrow 4.5\%}$	0.587	0.549	0.519	0.520	0.554	0.593	**0.621**$^{\uparrow 5.9\%}$
	syn	0.619	0.594	0.535	0.576	0.582	0.623	**0.659**$^{\uparrow 5.8\%}$	0.548	0.526	0.503	0.529	0.524	0.549	**0.577**$^{\uparrow 5.1\%}$

Table 1: The AUC results of training data detection across various models on the WikiMIA and StackMIAsub. In particular, the percentage data represent the minimum percentage performance improvement of our *PAC* method.

Generation Method					
Dataset	None	Neighbor	replace	delete	ours
WikiMIA	-4.2%	-7.5%	-8.0%	-7.2%	0.706
StackMIAsub	-5.4%	-12.7%	-6.9%	-6.3%	0.648
Metric Score					
Dataset	PPL	Zlib	Min	Max	ours
WikiMIA	-17.4%	-17.4%	-2.1%	-14.4%	0.706
StackMIAsub	-22.3%	-23.4%	-0.7%	-22.0%	0.648

Table 2: The AUC results on different generation methods and metric scores. 'replace' and 'delete' denote synonym replacement and random deletion, respectively. The scores not mentioned before include the log probability sum of tokens in high-probability regions.

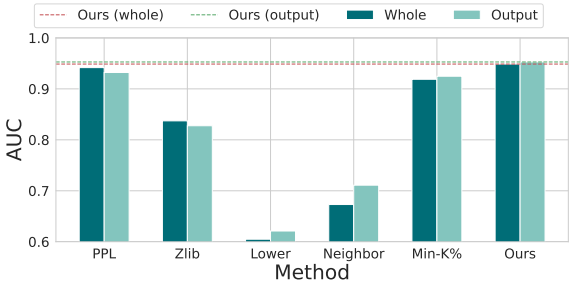

Figure 4: The AUC results in two-stage detection. 'whole' and 'output' represent two different settings of using the whole sample and the output part to detect.

works (Song and Shmatikov, 2019; Mahloujifar et al., 2021) have addressed this issue but are not extendable to a two-stage process. We select a recent clear fine-tuning dataset after contamination check, Platypus (Lee et al., 2023), to fine-tune the LLaMA-13B model under a 5 epoch set. To further simulate real-world scenarios, we conducted detection with both output portions and entire samples. As Figure 4 shows, *PAC* still exhibits excellent and stable performance even compared to the PPL score, which is directly equivalent to the training objective.

5.7 *PAC* as A Robust Threshold Interpreter

As mentioned in Section 3.1, the selection of threshold ϵ affects the final detection effectiveness. Thus, we focus on the threshold obtained in the scenario where the knowledge about all samples is limited. We randomly select 10%-50% (in 5% intervals) of the original dataset to form subsets. Then, following Lipton et al. (2014), we threshold *PAC* by maximizing the F1-score. As shown in Table 3, *PAC* consistently outperforms baselines by at least **3%** in accuracy. Simultaneously, the variance of the threshold ϵ obtained through subsets is relatively small (around 0.1), indicating that *PAC* requires only a small proportion of data to acquire a robust threshold, which is not limited by access-restricted

data.

Dataset	Metric	PPL	Zlib	Neighbor	Min-K%	Ours
WikiMIA	acc	0.59	0.57	0.59	0.58	**0.65**
	std	1.93	**0.01**	0.18	0.58	0.14*
StackMIAsub	acc	0.54	0.52	0.52	0.55	**0.58**
	std	1.32	**0.03**	0.18	0.25	0.09*
Mix	acc	0.55	0.53	0.52	0.55	**0.58**
	std	1.49	**0.03**	0.12	0.26	0.07*

Table 3: Results of threshold selection, where 'acc' and 'std' represent the accuracy and standard variance. The 'Mix' denotes a mixed data set of the others.

6 Case Study: Date Contamination

To further uncover the potential risks of existing LLMs (Large Language Models) through *PAC*, we selected two logical reasoning datasets, GSM8K (Cobbe et al., 2021) and AQuA (Garcia et al., 2020), and one ethical bias investigation dataset, TOXIGEN (Hartvigsen et al., 2022) on GPTs LLMs. As depicted in Table 4, both GPT-3 and the more advanced ChatGPT and GPT-4 exhibited varying degrees of contamination, reaching up to **91.4%** on davinci-002. Furthermore, all models unfortunately showed severe ethical bias data contamination in the training set (Figure 5). Based on the above, we call on the community to focus on finding solutions to the contamination problem to develop safer and more robust LLMs.

Model	GSM8K		AQuA		TOXIGEN		Avg
	Rate	Total	Rate	Total	Rate	Total	
davinci-002	95.3%	1319	89.8%	254	45.5%	178	89.4%
babbage-002	84.6%	1319	72.4%	254	33.7%	178	77.7%
gpt-3.5-turbo	82.0%	200	13.5%	200	5.06%	178	34.6%
GPT-4	64.0%	50	34.0%	50	6.7%	178	21.9%

Table 4: The cases of data contamination on GPTs. The results show both GPT-3 and the more advanced ChatGPT and GPT-4 exhibit varying degrees of contamination.

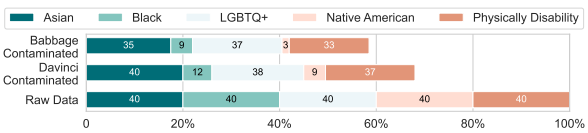

Figure 5: Bias data contamination cases of GPT-3 models. Cases are randomly selected from the TOXIGEN dataset.

7 Conclusion

We introduce Polarized Augment Calibration (*PAC*), a groundbreaking approach that expands the Membership Inference Attack (MIA) framework to detect training data in black-box LLMs. *PAC* unveils a new angle for MIA by utilizing confidence discrepancies across spatial data distributions and innovatively considering both distant and proximal probability regions to refine confidence metrics. This method is rigorously backed by theory and proven through comprehensive testing. We also present a novel detection technique for API-based black-box models using a proprietary probability tracking algorithm and launch StackMIA, a dataset aimed at overcoming the limitations of existing pre-trained data detection datasets. Applying *PAC* exposes widespread data contamination issues in even the most advanced LLMs, urging a communal effort towards addressing these challenges.

8 Limitations

While *PAC* shows promising results in detecting training data contamination in LLMs, its full potential is yet to be realized due to certain constraints. The limited availability of detailed training data information from LLMs providers restricts comprehensive validation across diverse models, underscoring the method's novelty yet implicating its untapped applicability. Additionally, the efficacy of *PAC* could be further enhanced with a more varied dataset, suggesting its adaptability and scope for refinement in varied LLMs. However, our current computational resources limit the extent of experiments, particularly on larger-scale LLMs, hinting at the method's scalability potential yet to be fully explored.

Acknowledgements

This work is supported by the Pioneer R&D Program of Zhejiang (No. 2024C01035), NSFC under Grants (No. 62206247), and the Fundamental Research Funds for the Central Universities (No. 226-2024-00049).

References

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

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

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. 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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

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, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284.

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

Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

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

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.

Noa Garcia, Chentao Ye, Zihua Liu, Qingtao Hu, Mayu Otani, Chenhui Chu, Yuta Nakashima, and Teruko Mitamura. 2020. A dataset and baselines for visual question answering on art. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 92–108. Springer.

Shahriar Golchin and Mihai Surdeanu. 2023. Time travel in llms: Tracing data contamination in large language models. *arXiv preprint arXiv:2308.08493*.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for implicit and adversarial hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. 2022. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv e-prints*, pages arXiv–2210.

Shotaro Ishihara. 2023. Training data extraction from pre-trained language models: A survey. *arXiv preprint arXiv:2305.16157*.

Bargav Jayaraman and David Evans. 2019. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1895–1912.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and William B Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069.

Petro Liashchynskyi and Pavlo Liashchynskyi. 2019. Grid search, random search, genetic algorithm: A big comparison for nas. *arXiv preprint arXiv:1912.06059*.

Zachary C Lipton, Charles Elkan, and B Narayanaswamy. 2014. Thresholding classifiers to maximize f1 score. *stat*, 1050:14.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165.

Saeed Mahloujifar, Huseyin A Inan, Melissa Chase, Esha Ghosh, and Marcello Hasegawa. 2021. Membership inference on word embedding and beyond. *arXiv preprint arXiv:2106.11384*.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*.

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 *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8332–8347.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Arvind Narayanan and Sayash Kapoor. 2023. Gpt-4 and professional benchmarks: the wrong answer to the wrong question. *AI Snake Oil*, 20:2023.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models.

Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. 2023. Tight auditing of differentially private machine learning. *arXiv preprint arXiv:2302.07956*.

Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018a. Comprehensive privacy analysis of deep learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, pages 1–15.

Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018b. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 634–646.

Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. 2021. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pages 866–882. IEEE.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, and Eneko Agirre. 2023. Did chatgpt cheat on your test?

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Network and Distributed Systems Security (NDSS) Symposium 2019*.

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. In *NeurIPS 2023 Workshop on Regulatable ML*.

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.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models.

Congzheng Song and Vitaly Shmatikov. 2019. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206.

Thomas Steinke, Milad Nasr, and Matthew Jagielski. 2023. Privacy auditing with one (1) training run. *arXiv preprint arXiv:2305.08846*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jonathan Tow, Marco Bellagente, Dakota Mahan, and Carlos Riquelme Ruiz. 2023. Stablelm-3b-4e1t.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. 2021. On the importance of difficulty calibration in membership inference attacks. In *International Conference on Learning Representations*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388.

Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2023. " according to..." prompting language models improves quoting from pre-training data. *arXiv preprint arXiv:2305.13252*.

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.

Wentao Ye, Mingfeng Ou, Tianyi Li, Xuetao Ma, Yifan Yanggong, Sai Wu, Jie Fu, Gang Chen, Junbo Zhao, et al. 2023. Assessing hidden risks of llms: An empirical study on robustness, consistency, and credibility. *arXiv preprint arXiv:2305.10235*.

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.

Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. 2020. Analyzing information leakage of updates to natural language models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 363–375.

Elias A Zerhouni and Elizabeth G Nabel. 2008. Protecting aggregate genomic data. *Science*, 322(5898):44–44.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv e-prints*, pages arXiv–2205.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv e-prints*, pages arXiv–2303.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't make your llm an evaluation benchmark cheater. *arXiv e-prints*, pages arXiv–2311.

A Details of StackMIA

A.1 Stack Exchange

The Stack Exchange Data Dump contains user-contributed content on the Stack Exchange network. It is one of the largest publicly available repositories of question-answer pairs and covers a wide range of subjects –from programming to gardening, to Buddhism. Many large language models therefore include this dataset in their training data to enrich the training data and improve the model's ability to answer questions in different domains.

A.2 StackMIAsub

We follow Shi et al. (2023) to divide all the samples we collected into 4 categories based on length, ranging from 32 words to 256 words. The specific composition of StackMIA is shown in the Table 5.

Length	32	64	128	256	Total
Member	1596	1740	680	90	4106
Non-member	1615	1740	720	86	4161

Table 5: The samples composition of StackMIA dataset.

A.3 Synonymous Rewritten Data

As mentioned in Section 3.2, we rewrite the original samples to simulate approximate memorization scenarios by prompting GPT-3.5-turbo API. Part of the prompts we used to rewrite the sentence are listed as follows:

- "Rewrite the sentence with the smallest possible margin, keeping the same semantics and do not complete anything, ensuring that BLEU > 0.75 before and after the change: '*text*' "
- "Slightly rewrite the following sentence without changing the sentence structure, and do not complete any sentence: '*text*' "
- "Randomly replace 2 words in the giving sentence with synonyms: '*text*' "
- "Replace 5 percent of the prepositions with a synonym in the giving sentence: '*text*' "

The case comparison of the samples before and after the synonymous transformation with ChatGPT is listed in Table 7.

Length	32	64	128	256	Total
Member	387	284	139	51	861
Non-member	389	258	111	31	789

Table 6: The samples composition of WikiMIA Dataset.

B Details of WikiMIA

WikiMIA is a benchmark for MIA((Shi et al., 2023)), with data sourced from Wikipedia. For non-member data, the dataset collects recent event pages using January 1, 2023, as the cutoff date. For member data, the dataset collects articles before 2017. The specific composition of WikiMIA is shown in the Table 6.

C A Simple Proof of PAC method

Here we provide a mathematical proof why our method can effectively distinguishes members and non-members.

As mentioned in Equation 5, given a sample z, the Polarized Distance can be briefly represented as the difference between the two terms T_1 and T_2:

$$\begin{aligned}\mathrm{T}_1 &= \frac{1}{K_1} \sum_{u_i \in \mathrm{MAX}(z,k_1)} log f_\theta(u_i|u_1,\cdots,u_{i-1}) \\ \mathrm{T}_2 &= \frac{1}{K_2} \sum_{u_i \in \mathrm{MIN}(z,k_2)} log f_\theta(u_i|u_1,\cdots,u_{i-1})\end{aligned} \tag{9}$$

Each term can be scaled to the following inequality:

$$\begin{aligned}\frac{|z|}{K_1}\mathrm{T}_1 &\geqslant \mathbb{E}_{u\in z}[\log f_\theta(u|\cdot)] \\ \frac{|z|}{K_2}\mathrm{T}_2 &\leqslant \mathbb{E}_{u\in z}[\log f_\theta(u|\cdot)]\end{aligned} \tag{10}$$

where $\mathbb{E}$ function denotes the expectation function and $\cdot$ denotes the prefix tokens sequence. Then the MinMax Distance can be represented as:

$$0 \geqslant \mathcal{L}_M = \mathrm{T}_1 - \mathrm{T}_2 \geqslant \frac{K_1-K_2}{|z|}\mathbb{E}_{u\in z}[\log f_\theta(u|\cdot)] \tag{11}$$

Then Equation 6 can be further converted to:

$$\begin{aligned}&\frac{K_1-K_2}{|z|}\mathbb{E}_{u\in z}[\log f_\theta(u|\cdot)] \leqslant \\ &\mathcal{L}_M(z) - \mathcal{L}_M(\sigma_m(z)) \leqslant \\ &\frac{K_2-K_1}{|\sigma_m(z)|}\mathbb{E}_{u\in \sigma_m(z)}[\log f_\theta(u|\cdot)]\end{aligned} \tag{12}$$

Since the coefficient is constant, the threshold interval of the final calculated indicator can be equivalently expressed as:

$$\mathbb{E}_{u\in z}[\log f_\theta(u|\cdot)] + \mathbb{E}_{u\in \sigma_m(z)}[\log f_\theta(u|\cdot)] \tag{13}$$

Among them, the former term can be considered as a variant of the LLM training objective. Therefore, the value of Formula 13 will change significantly as z is fitted as member data. Moreover, there exists a positive correlation between the posterior term and the forward tendency according to Jin et al. (2020). This means that the threshold intervals of members and non-members are more sparse than solely using the probability of z directly. Therefore, our indicator significantly captures differences between members and non-members.

D Proof of Probability Extraction

Assume that the logit output of the model $f_\theta(u_i|\cdot)$ is $l_1, \cdots, l_N$, and then the log-probability of u_i can be represented as:

$$\log f_\theta(u_i|\cdot) = \log \frac{e^{l_i}}{\sum_{j=1}^{M} e^{l_j}} = l_i - \log \sum_{j=1}^{N} e^{l_j} \tag{14}$$

Then, the log-probability $\log f'_\theta(u_i|\cdot)$ after adding a fixed bias γ_i to the logit of u_i can be calculated as:

$$\begin{aligned} \log f'_\theta(u_i|\cdot) &\approx \log \frac{e^{l_i+\gamma_i}}{\sum_{j=1}^{N} e^{l_j}} \\ &= (l_i + \gamma_i) - \log \sum_{j=1}^{N} e^{l_j} \end{aligned} \tag{15}$$

Then the original log-probability can be calculated as:

$$\log f_\theta(u_i|\cdot) \approx \log f'_\theta(u_i|\cdot) - \gamma_i \tag{16}$$

E Baseline Details

E.1 Reference model comparison

Reference model-based methods target at training reference models in the same manner as the target model (e.g., on the shadow data sampled from the same underlying pretraining data distribution). The raw score of the original samples can be calibrated with the average of the score in these reference models. Due to the high cost of strictly training a shadow LLM, we follow Carlini et al. (2021) to choose a much smaller model trained on the same underlying dataset. Specifically, we choose LLaMA-7B as the reference model of LLaMA-13B, LLaMA2-7B for LLaMA2-13B, GPT-Neo-125M for GPT-Neo-2.7B, OPT-125M for OPT-6.7B, Pythia-70M for Pythia-6.9B, StableLM 3B for StableLM-7B. Specially, for those LLMs without smaller models in the series, we use an approximate model trained on the mentioned same-dataset or distribution in their official statement, including GPT-Neo-125M for GPT-J-6B both trained on the Pile (Gao et al.), GPT-Neo-125M for Falcon-7B trained on the Pile and GPT2-124M (Radford et al.) for two OpenAI base models (Davinci-002 and Babbage-002).

E.2 Neighborhood Attack via Neighbourhood Comparison

This method is proposed by Mattern et al. (2023). The main idea is to compare the neighbors' losses and those of the original sample under the target model by computing their differences.

In our replication process, we followed the method on official Github [8] to select key hyperparameters. Specifically, We randomly mask 30 percent of the tokens in the original sample with a span length of 2 to generate 25 masked sentences from the original sample and then use the T5-3b mask-filling model to generate 25 neighbors.

To evaluate the score of a sample, we use the formula as follows:

$$\frac{\mathcal{L}(x) - \sum^{N} \frac{\mathcal{L}(\tilde{x})}{N}}{\sigma}$$

where σ is the standard deviation of the neighbours' losses.

E.3 Min-K%

The Min-K% method, proposed by Shi et al. (2023), is quite straightforward. It builds on the hypothesis that a non-member example is more likely to include a few outlier words with high negative log-likelihood (or low probability), while a member example is less likely to include words with high negative log-likelihood. Specifically, Min-K% is calculated as :

$$\text{model}(x) = \frac{1}{E} \sum_{x_i \in \text{Min-K\%}(x)} \log p(x_i|x_1, ..., x_{i-1}). \tag{17}$$

where $x = x_1, x_2, ..., x_N$ is the tokens in the sentences, while $\log p(x_i|x_1, ..., x_{i-1})$ is the log-likelihood of a token, x_i.

In particular, we follow the original paper to set $K = 20$ for detection in experiments.

F Descriptions of AUC score

To evaluate with AUC score, we first plot the ROC curve through the True Positive Rate (TPR) and False Positive Rate(FPR). The ROC curve is used to

[8] https://github.com/mireshghallah/neighborhood-curvature-mia

plot TPR versus FPR using different classification thresholds. Lowering the classification threshold causes more categories of items to be classified as positive, thus increasing the number of false positives and true examples. AUC is then defined as the Area Under the ROC curve, providing an aggregate measure of the effect of all possible classification thresholds.

G Sentiment Analysis

We further conducted syntax analysis experiments on both member and non-member samples. The specific results are shown in Figure 6.

Figure 6: Sentiment analysis on WikiMIA & StackMIA datasets. The X-axis represents the prediction score given by AMD, and the Y-axis represents the sentiment analysis score, with higher scores meaning positive and lower scores meaning negative.

H More Experiments Demonstrations

The visualizing AUC results of training data detection across various models with different methods are shown in Figure 7 & 8.

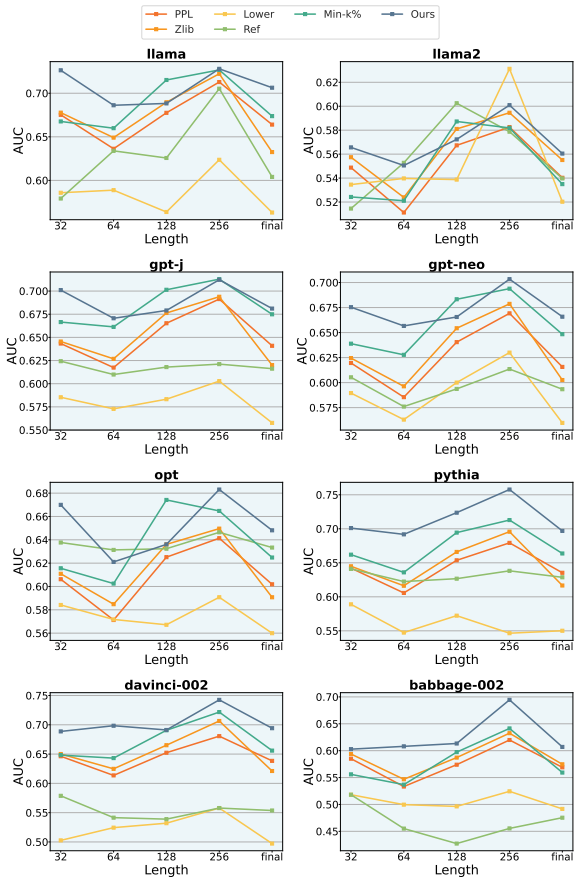

Figure 7: The AUC results of training data detection across various models with different methods on the WikiMIA dataset.

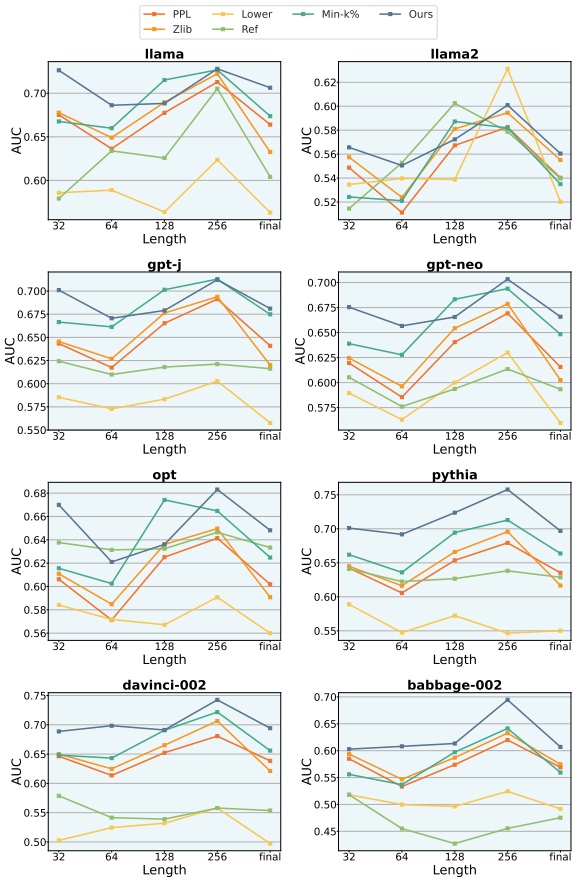

Figure 8: The AUC results of training data detection across various models with different methods on the StackMIAsub dataset.

ori	Density Functional Theory (DFT) is formulated to obtain ground state properties of atoms, molecules and condensed matter. However, why is DFT not able to predict the exact band gaps of semiconductors and insulators? Does it mean that the band gaps of semiconductors and insulators are not the ground states?
syn	Why is it that Density Functional Theory (DFT) cannot accurately predict the precise band gaps of semiconductors and insulators, even though it is designed to determine the ground state properties of atoms, molecules, and condensed matter? Does this imply that the band gaps of semiconductors and insulators are not considered as ground states?
ori	I am currently studying Electrical & Electronic Engineering. I wish to pursue Quantum Mechanics or Quantum Computing as my research subject. Is it possible for me to do my M.Tech. and then pursue my research subject? What are the prerequisites for studying these subjects? I would be grateful if you could help me.
syn	I am presently studying Electrical & Electronic Engineering. I desire to pursue Quantum Mechanics or Quantum Computing as my research topic. Is it feasible for me to do my M.Tech. and then pursue my research topic? What are the requirements for studying these subjects? I would be thankful if you could assist me.
ori	How is the meaning of a sentences affected by chosing one of those words? For instance, what's the different between The screech cicadas reverberated through the forest. and The screech cicadas reverberated throughout the forest.
syn	How does the choice of one of those words affect the meaning of a sentence? For example, what is the difference between T̈he screech cicadas reverberated through the forest.änd T̈he screech cicadas reverberated throughout the forest.ï̈
ori	The majority of definitions give the same meaning - P̈andora's boxïs a synonym for ä source of extensive but unforeseen troubles or problems.Äre there any other metaphors or phrases with the same meaning?
syn	Do any other metaphors or phrases convey the same meaning as the majority of definitions, which state that P̈andora's boxïs synonymous with ä source of extensive but unforeseen troubles or problemsï̈
ori	The majority of definitions give the same meaning - P̈andora's boxïs a synonym for ä source of extensive but unforeseen troubles or problems.Äre there any other metaphors or phrases with the same meaning?
syn	Do any other metaphors or phrases convey the same meaning as the majority of definitions, which state that P̈andora's boxïs synonymous with ä source of extensive but unforeseen troubles or problemsï̈"

Table 7: The cases before and after are synonymous rewritten with ChatGPT. The listed cases are selected from the member data with a length between 32 and 64.