Automated Detection of Pre-training Text in Black-box LLMs*

Ruihan Hu[1], **Yu-Ming Shang**[1], **Jiankun Peng**[1], **Wei Luo**[1], **Yazhe Wang**[2] and **Xi Zhang**[1†]
[1]Key Laboratory of Trustworthy Distributed Computing and Service (MoE),
Beijing University of Posts and Telecommunications, China
[2]Zhongguancun Laboratory, China
{gloria-1019, shangym, projankt, luowei}@bupt.edu.cn, wangyz@zgclab.edu.cn, zhangx@bupt.edu.cn

Abstract

Detecting whether a given text is a member of the pre-training data of Large Language Models (LLMs) is crucial for ensuring data privacy and copyright protection. Most existing methods rely on the LLM's hidden information (e.g., model parameters or token probabilities), making them ineffective in the black-box setting, where only input and output texts are accessible. Although some methods have been proposed for the black-box setting, they rely on massive manual efforts such as designing complicated questions or instructions. To address these issues, we propose VeilProbe, the first framework for automatically detecting LLMs' pre-training texts in a black-box setting without human intervention. VeilProbe utilizes a sequence-to-sequence mapping model to infer the latent mapping feature between the input text and the corresponding output suffix generated by the LLM. Then it performs the key token perturbations to obtain more distinguishable membership features. Additionally, considering real-world scenarios where the ground-truth training text samples are limited, a prototype-based membership classifier is introduced to alleviate the overfitting issue. Extensive evaluations on three widely used datasets demonstrate that our framework is effective and superior in the black-box setting.

1 Introduction

The extensive corpus used during the pre-training phase is a major factor behind the extraordinary performance of Large Language Models (LLMs) [Brown *et al.*, 2020]. In fact, some corpora potentially include sensitive information, such as copyrighted materials. This could give rise to legal disputes, such as The New York Times lawsuit against OpenAI [Grynbaum and Mac, 2023] for unauthorized use of its articles in training LLMs. Pre-training data detection task in LLMs aims to determine whether a given material is included in the target LLM's (*i.e.*, the LLM to be detected) pre-training corpus [Shi *et al.*, 2024]. It has become a research hotspot due to its important value in addressing issues such as detecting data contamination [Oren *et al.*, 2023; Ye *et al.*, 2024] and protecting the rights of copyright holders [Karamolegkou *et al.*, 2023; Chang *et al.*, 2023].

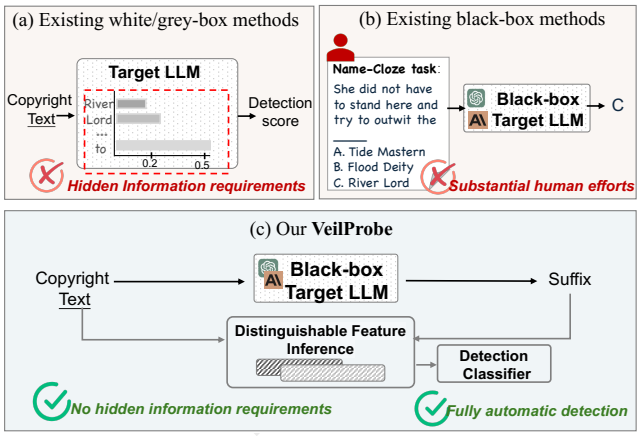


Figure 1: The comparation between existing methods and **VeilProbe (ours)**. (a) Existing white/grey-box methods should rely on hidden information of LLMs to calculate distinguishable scores for detection. (b) Exiting black-box methods requires substantial human efforts to design specific tasks for each text. (c) Our VeilProbe relies solely on text-to-suffix pairs to achieve automatic detection.

Pre-training detection methods typically rely on the target LLM's distinct output behavior when the given pre-training texts are used as input. According to the detector's access levels, they can be categorized as the white-box [Wang *et al.*, 2024], grey-box [Shi *et al.*, 2024; Zhang *et al.*, 2024b], and black-box settings [Duarte *et al.*, 2024; Chang *et al.*, 2023]. The methods in white-box and grey-box settings, as illustrated in Fig. 1 (a), generally follow a paradigm that calculates a distinguishable score based on the statistical characteristics of token probabilities or hidden parameters [Carlini *et al.*, 2021; Shi *et al.*, 2024; Zhang *et al.*, 2024b]. For instance, Min-K% Prob [Shi *et al.*, 2024] uses the sum of the lowest token log probabilities as the score. DC-PDD [Zhang *et al.*, 2024b] introduces a calibrated score based on the cross-entropy between token probabilities and reference corpus frequencies. Nevertheless, advanced LLMs

*Code: `github.com/STAIR-BUPT/VeilProbe`
†Corresponding Author

like ChatGPT and Claude are often in the black-box settings, where only the input and output texts are accessible. White-box and grey-box methods are infeasible in this scenario.

Existing black-box pre-training data detection methods typically enable the target LLM to perform complicated tasks such as question-answering tasks [Karamolegkou *et al.*, 2023], multiple-choice questions [Duarte *et al.*, 2024], and name-cloze tasks [Chang *et al.*, 2023]. The LLM's responses distinguish between pre-training texts and non-training texts. Fig. 1 (b) shows an example that the name-cloze task requires deciding where to place blanks and creating customized options. Yet, it requires handcrafted instructions for each specific input, which is expensive and time-consuming and cannot adapt to real-world large-scale pre-training text detection.

The purpose of this study is to detect texts in the black-box setting without human involvement. One natural idea is to infer the membership distinguishable features and use these features to train a membership classifier to identify pre-training texts, as shown in Fig. 1 (c). However, it is non-trivial due to the following two challenges. (1) The commonly used intuitive indicators [Dong *et al.*, 2024; Abbasi-Yadkori *et al.*, 2024], which are directly computed based on the input and output texts, such as semantic shifts or consistency of output texts, have been proven to be inadequate to distinguish members (see our experiments in Appendix C.1); (2) The number of ground-truth pre-training samples is quite limited as the LLM providers usually do not disclose [Karamolegkou *et al.*, 2023; Shi *et al.*, 2024; Chang *et al.*, 2023]. This would result in overfitting issues when training the membership classifier.

To tackle the above challenges, we propose a novel pre-training text detection framework **VeilProbe**, consisting of two main parts: the membership feature inference module, and the prototype-based classifier. To capture the distinguishable features between training and non-training texts, the inference module first uses a sequence-to-sequence mapping model to simulate the mapping pattern of how the LLM completes a given text to its suffix. Additionally, inspired by [Liu *et al.*, 2024], it integrates the perturbation calibration features by perturbing the key tokens in the input text to enrich the distinguishable features. Using these features, a prototype-based classifier [Snell *et al.*, 2017] integrated with Information Bottleneck (IB) [Tishby and Zaslavsky, 2015] is proposed for the detection of pre-training texts. This classifier is capable of filtering out irrelevant features. Moreover, it allows for effective classification using only a limited number of ground-truth training instances. Our contributions are as follows:

- We propose a novel framework named **VeilProbe**. To the best of our knowledge, we are the first to automatically detect pre-training text in the black-box LLMs without human involvement.
- We propose a sequence-to-sequence model aiming to capture the intricate Text-to-Suffix pattern and infer the features that can distinguish membership. Additionally, we design a prototype-based classifier, which is capable of identifying the membership training data by leveraging merely a limited number of known samples.
- We compare the performance of VeilProbe to a wide range of baselines. It shows state-of-art results on three widely used datasets and outperforms the existing methods by a good margin.

2 Related Work

2.1 Membership Inference Attack in LLMs

Membership Inference Attack (MIA) [Carlini *et al.*, 2022; Shokri *et al.*, 2017] aims to identify whether a specific data point is in the model's training set. The concept of MIA, which was initially applied in the Computer Vision domain [Shokri *et al.*, 2017], has recently been extended to the Natural Language Processing tasks [Carlini *et al.*, 2021; Ye *et al.*, 2022; Duan *et al.*, 2024]. MIA on LLMs can serve as a technique for several relevant tasks, such as Data Extraction Attacks [Carlini *et al.*, 2021; Yu *et al.*, 2023] and Personally Identifiable Information Attacks [Lukas *et al.*, 2023; Shi *et al.*, 2024; Kim *et al.*, 2023], data contamination [Oren *et al.*, 2023; Golchin and Surdeanu, 2024] and finetuning data detection [Mireshghallah *et al.*, ; Song and Shmatikov, 2019]. Our works focus on an instance of MIA, pre-training data detection, a task that has become a hot issue in recent research.

2.2 Pre-training Data Detection

According to the access level for detectors, there are three categories: the white-box, grey-box, and black-box settings. As the settings become increasingly strict, the information available to the detector diminishes, making the detection more challenging. Most studies focus on the grey-box setting, where the token probability distribution is accessible. Most of them [Carlini *et al.*, 2021; Mattern *et al.*, 2023; Shi *et al.*, 2024; Zhang *et al.*, 2024a; Zhang *et al.*, 2024b] typically rely on heuristic assumptions to calculate a membership score for identifying pre-training texts. Recently, some studies [Maini *et al.*, 2024; Puerto *et al.*, 2024]selectively combine the membership scores as the aggregated feature. We focus on the more challenging black-box setting. Some previous works [Duarte *et al.*, 2024; Chang *et al.*, 2023]in such settings usually design complicated tasks to identify pre-training texts. Unlike previous work, our work focuses on automating the detection process in the black-box setting without human intervention.

3 Problem Statement

Formally, given a text $s \in S$, where the S denotes the set of texts to be detected, and a target LLM Θ with pre-training data D_θ, we aim to determine whether s is a member in the pre-training dataset D_θ of Θ. Building upon previous work [Shi *et al.*, 2024; Zhang *et al.*, 2024b], the task is defined as follows:

$$\mathcal{A}(s|\Theta, \mathcal{K}) \rightarrow \{0, 1\}, \tag{1}$$

where $\mathcal{A}$ denotes the detection algorithm, and $\mathcal{K}$ represents the information available to the detector. In the black-box setting, the detector can only access the input and output texts generated by Θ. We assume a few ground-truth pre-training and non-training text samples are available, which is in line with the practical scenarios. In particular, for

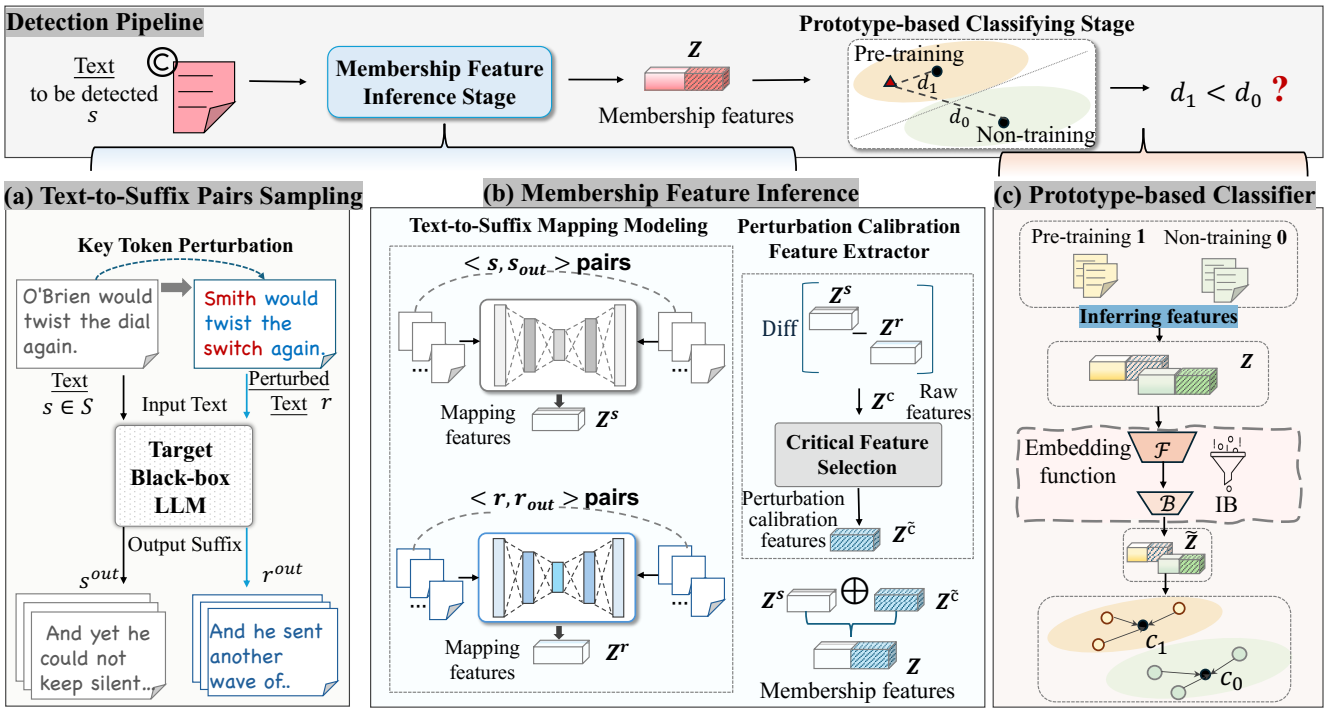


Figure 2: Overview of **VeilProbe**. (a) The text-to-suffix sampling module generates text-to-suffix pairs; (b) The membership feature inference module infers membership features with a sequence-to-sequence model based on the above pairs; (c) The prototype-based classifier is trained based on the features from ground-truth samples, and then the pre-training and non-training prototypes are constructed. The first two modules prepare the membership feature inference stage for the texts to be detected, and the third one trains a membership classifier for detection.

the pre-training samples, a few classic popular published books, papers, and Wikipedia pages are well known to be included in the pre-training corpus [Chang *et al.*, 2023; Karamolegkou *et al.*, 2023; Zhu *et al.*, 2015]. Moreover, there are also some unpublished manuscripts from some writers that can be the non-training samples, which is consistent with the setting in previous studies such as [Maini *et al.*, 2024; Puerto *et al.*, 2024].

4 Method

Our proposed **VeilProbe** framework is illustrated in Figure 2, which is composed of three modules: (1) For preprocessing, the text-to-suffix pairs sampling module is proposed to sample the completion suffix s^{out} of each input text $s \in S$ by requesting the target LLM response. Moreover, we perturb the key tokens of s to generate r, which is then fed into LLMs to obtain the corresponding suffix r^{out}. (2) Based on multiple $\langle s, s^{\text{out}} \rangle$ and $\langle r, r^{\text{out}} \rangle$ pairs, the membership feature inference module is devised to infer the latent membership feature $\boldsymbol{Z}$. (3) The prototype-based classifier is trained with the inferred membership features to obtain the prototypes for pre-training and non-training data. In the detection pipeline, given a text s to be detected, if its membership feature is closer to the pre-training prototype, it can be inferred that s is a member of the target LLM's pre-training data.

4.1 Text-to-Suffix Pairs Sampling

As shown in Figure 2 (a), for preprocessing, the sampling stage aims to generate a set of text-to-suffix pairs from the target black-box LLM, including text-to-suffix pairs $\langle s, s^{\text{out}} \rangle$ and perturbed text-to-suffix pairs $\langle r, r^{\text{out}} \rangle$.

Text-to-Suffix Pair Generation. The autoregressive completion task is employed to align with the training patterns of the corpus during the pre-training phase. To be specific, without additional instructions, each text s serves as a prefix, and the target black-box LLM Θ is requested to complete the sentence based on s. The target LLM autoregressively generates the subsequent tokens until the maximum generation length is reached. A pair of the text s and its corresponding response s^{out} is denoted as $\langle s, s^{\text{out}} \rangle$. More than one suffix is generated for each text to reduce the randomness in the responses. Similarly, the corresponding suffix r^{out} of the perturbed text r is generated in the same autoregressive way. The following will elaborate on how to obtain the perturbed r.

Key Token Perturbation. We carry out perturbations on the key tokens within the original input s to generate r. The key tokens are those that exert substantial influence on the generation of the suffix text. Inspired by [Li *et al.*, 2024], we employ a feature attribution technique to identify key tokens. However, directly applying feature attribution analysis to the black-box LLM is non-trivial. Most of these methods depend on hidden information (prediction or token probability distributions), which is unsuitable for the black-box scenario.

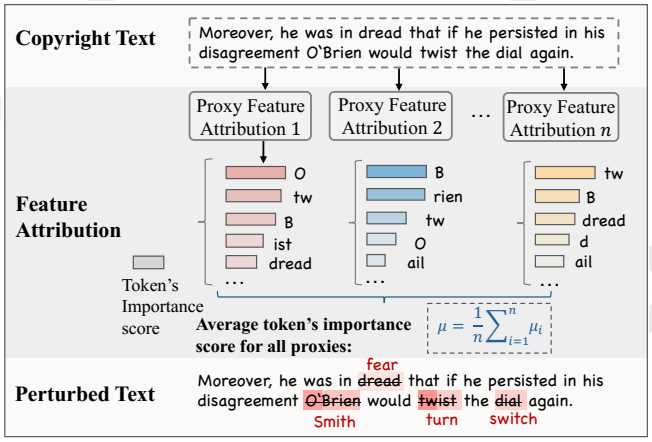


Figure 3: The pipeline of key token perturbation module.

To address the above issue, a perturbation-based feature attribution method is introduced, as illustrated in Figure 3. We use multiple small-size open-source LLMs (preferably from the same family) as proxy models to assist in selecting the key tokens. It is observed that when using LLMs from the same family as proxy LLMs, the feature attribution technique tends to select a similar set of informative tokens in a text. For instance, GPT-2 can serve as a proxy LLM for ChatGPT. (See Appendix C.2 for detailed analysis). Then the importance scores obtained for each token across multiple proxy LLMs are averaged to obtain the final importance score μ. Based on these scores, top-$\gamma\%$ tokens are selected and perturbed with synonyms to produce the perturbed text r.

4.2 Membership Feature Inference

This module is designed to infer membership features based on the $\langle s, s^{\text{out}} \rangle$ and $\langle r, r^{\text{out}} \rangle$ pairs obtained from the above module.

Text-to-Suffix Mapping Modeling. Since LLMs generate outputs one by one in an autoregressive manner, the input-output mapping f generally follows a sequence-to-sequence pattern [Sutskever *et al.*, 2014; Vaswani *et al.*, 2023]. To capture such a pattern, a sequence-to-sequence mapping model based on the transformer architecture [Vaswani *et al.*, 2023] is proposed to model the mapping pattern from the input text to its suffix. Specifically, this mapping model is trained on $\langle s, s^{\text{out}} \rangle$ pairs to learn the input-output dependency, that is, $f : s \rightarrow s^{\text{out}}$. The training objective is to maximize the conditional probability of the output sequence s^{out} given the input sequence s as the prefix, which is expressed as:

$$P(s^{\text{out}}|s) = \prod_{t=1}^{T} P(w_t^{\text{out}}|w_1^{\text{out}}, w_2^{\text{out}}, \ldots, w_{t-1}^{\text{out}}, s), \quad (2)$$

where T denotes the max length of the output sequence s^{out}, and each w_t^{out} represents the token generated at time step t. The model learns to predict w_t^{out} based on the previously generated tokens $w_1^{\text{out}}, \ldots, w_{t-1}^{\text{out}}$ and the input sequence s. Upon completion of the training process, once a text input is provided to the sequence-to-sequence model, the hidden states of the model are extracted as the text-to-suffix mapping feature $\boldsymbol{Z}^{\text{s}}$. Similarly, the perturbed mapping features $\boldsymbol{Z}^{\text{r}}$ can be obtained based on $\langle r, r^{\text{out}} \rangle$ pairs.

The obtained mapping feature can function as a part of the membership distinguishable features to distinguish between pre-training and non-training texts. The fundamental concept is based on previous research [Wang *et al.*, 2024] which has shown that the hidden states in the internal layers of a white-box LLM can tell the difference between the pre-training texts and non-training texts.

Perturbation Calibration Feature Extractor. In addition to the hidden states obtained from the aforementioned mapping model, we further explore the perturbation calibration feature as a complement. The intuition is based on the observation that the robustness of the target model exhibits marked differences when perturbed pre-training samples and non-training samples are employed as inputs [Liu *et al.*, 2024; Mattern *et al.*, 2023].

Specifically, as shown in Figure 2 (b), we first obtain the raw perturbation feature $\boldsymbol{Z}^{\text{c}}$, which is the difference between the original $\boldsymbol{Z}^{\text{s}}$ and perturbed mapping features $\boldsymbol{Z}^{\text{r}}$. Since the perturbation operations would probably induce noisy information [Liu *et al.*, 2024; Hooker *et al.*, 2019] for identifying members, it is crucial to select the most critical features to infer membership status. The significance test [WELCH, 1947] is applied to filter critical features. We formulate the null hypothesis (H_0) and the alternative hypothesis (H_1) for the values in each dimension of the feature with respect to their impact on the target variable (i.e., membership status Y). The hypotheses are defined as follows:

$$H_{0i} : \Pr(Y = 1 \mid z_i^{\text{c}} \in G_1) = \Pr(Y = 1 \mid z_i^{\text{c}} \in G_0), \quad (3)$$
$$H_{1i} : \Pr(Y = 1 \mid z_i^{\text{c}} \in G_1) \neq \Pr(Y = 1 \mid z_i^{\text{c}} \in G_0), \quad (4)$$

where G_1, and G_0 represent the ground-truth pre-training and non-training samples, respectively. z_i^{c} represents the i-th dimension of the feature vector $\boldsymbol{Z}^{\text{c}}$. Then the p-value of each dimension of the feature vector is obtained. A lower p-value indicates strong evidence against the null hypothesis, suggesting that the pre-training and non-training samples are distinct in the i-th dimension, supporting the alternative hypothesis. We then retain the critical dimensions and set the others to zero. The processed features then serve as the final perturbation calibration feature $\boldsymbol{Z}^{\widetilde{\text{c}}}$.

Consequently, the membership feature of s is the concatenation of the mapping feature $\boldsymbol{Z}^{\text{s}}$ and the perturbation calibration feature $\boldsymbol{Z}^{\widetilde{\text{c}}}$, which is defined as:

$$\boldsymbol{Z} = \boldsymbol{Z}^{\text{s}} \oplus \boldsymbol{Z}^{\widetilde{\text{c}}}. \quad (5)$$

4.3 Prototype-based Classifier

Based on the membership features, we propose a prototype-based classifier to distinguish between pre-training and non-training texts. We choose this classifier as a backbone because it works well even with limited labeled data.

Prototype Constructing. The prototype-based classifier constructs a metric space through learning [Snell *et al.*, 2017]. In this space, samples belonging to the same class are proximal to each other, whereas samples from different classes are distantly separated.

Each class is represented by a prototype, which is the center of its samples in this space. As shown in Figure 2 (c), based on the membership features, we aim to construct the pre-training prototype and the non-training prototype. Specifically, the embedding function $\mathcal{F}$ is trained following the approach of Prototypical Networks [Snell *et al.*, 2017] to mitigate overfitting issues caused by the limited labeled data. The training episodes are constructed by randomly sampling a subset of samples from the ground-truth set G to form the support set U, while a separate subset of the remaining samples serves as query points.

Formally, we define the support set as $U = \{(\boldsymbol{Z}_1, Y_1), \ldots, (\boldsymbol{Z}_m, Y_m)\}$, where each $\boldsymbol{Z}_i \in \mathbb{R}^D$ represents the D-dimensional feature vector of an sample, and $Y_i \in \{0, 1\}$ is the corresponding label. U_Y represents the set of samples labeled with pre-training (U_1) or non-training (U_0). The objective is to compute an K-dimensional representation, denoted as $c_Y \in \mathbb{R}^K$, where c_0 represents the non-training prototype and c_1 represents the training prototype. These prototypes can be obtained using an embedding function $\mathcal{F}_\phi : \mathbb{R}^D \to \mathbb{R}^K$ with learnable parameters ϕ. Each prototype is defined as the mean vector of the embedded representations of the support samples belonging to their set:

$$c_Y = \frac{1}{|U_Y|} \sum_{(\boldsymbol{Z}_i, Y_i) \in U_Y} \mathcal{F}_\phi(\boldsymbol{Z}_i). \tag{6}$$

We define the probability distribution over class labels for a query point $\boldsymbol{Z}$ using a softmax function, which is based on the distances between the query's embedding and the class prototypes. The probability of classifying $\boldsymbol{Z}$ as belonging to class Y is given by:

$$p_\phi(y = Y \mid \boldsymbol{Z}) = \frac{\exp(-d(\mathcal{F}_\phi(\boldsymbol{Z}), c_Y))}{\sum_{Y'} \exp(-d(\mathcal{F}_\phi(\boldsymbol{Z}), c_{Y'}))}, \tag{7}$$

where $d(\mathcal{F}_\phi(\boldsymbol{Z}), c_Y)$ represents the distance between the query embedding $\mathcal{F}_\phi(\boldsymbol{Z})$ and the prototype c_Y of class Y, and the distance function d is squared Euclidean distance. The learning process minimizes the negative log-likelihood of the true class label via Stochastic Gradient Descent.

Information Bottleneck Denoising. To further filter out redundant features in $\boldsymbol{Z}$, we integrate the Information Bottleneck technique (IB) into the prototype computation. This technique aims to extract a compact representation that retains the information relevant to Y while eliminating redundant information from the input feature $\boldsymbol{Z}$. Specifically, we construct a compressed representation $\tilde{\boldsymbol{Z}}$ by optimizing the following objective:

$$\min \mathcal{L}_{\text{IB}} = I(\boldsymbol{Z}, \tilde{\boldsymbol{Z}}) - \beta I(\tilde{\boldsymbol{Z}}; Y), \tag{8}$$

where $I(\cdot)$ represents the mutual information, $\boldsymbol{Z}$ represents the input feature, Y is the ground-truth label, and β balances the trade-off between minimizing redundancy and retaining relevance to Y.

We then use the compressed representation $\tilde{\boldsymbol{Z}}$ as the feature to detect the pre-training text s. That is, the detection score δ is defined as:

$$\delta = d(\tilde{\boldsymbol{Z}}, c_0) - d(\tilde{\boldsymbol{Z}}, c_1), \tag{9}$$

Datasets	Len.	#Samples	Resources
WikiMIA [Shi *et al.*, 2024]	32	776	Wikipedia events
	64	542	
	128	250	
	256	82	
BookTection [Duarte *et al.*, 2024]	64	5472	165 Bestselling books
	128	5494	
	256	5447	
arXivTection [Duarte *et al.*, 2024]	128	1549	50 arXiv papers

Table 1: The statistics of three datasets.

where d represents the distance metric, c_1 represents the pre-training prototype, and c_0 represents the non-training prototype. If $\tilde{\boldsymbol{Z}}$ is closer to the c_1 than c_0, it can be inferred that the text s is a member in the pre-training data of the target LLM.

5 Experiments

5.1 Experimental settings

Datasets. The statistics of three datasets are shown in Table 1. **WikiMIA** [Shi *et al.*, 2024] consists of Wikipedia event snippets. **BookTection** [Duarte *et al.*, 2024] is a widely adopted dataset that contains 165 copyrighted books, which is expanded based on BookMIA [Shi *et al.*, 2024]. **arXivTection** [Duarte *et al.*, 2024] includes classic papers from arXiv.

Target LLMs. Following [Shi *et al.*, 2024], for WikiMIA, we select Pythia-6.9B [Biderman *et al.*, 2023], LLaMA-13B [Touvron *et al.*, 2023a], and GPT-NeoX-20B [Black *et al.*, 2022] as our target LLMs. Following [Duarte *et al.*, 2024], for BookTection, Mistral-7B [Jiang *et al.*, 2023], LLaMA2-7B [Touvron *et al.*, 2023b] and ChatGPT (gpt-3.5-instruct) are selected as the target LLMs. For arXivTection, Mistral-7B [Jiang *et al.*, 2023], LLaMA2-13B [Touvron *et al.*, 2023b], and Claude 2.1 are selected as our target LLMs.

Evaluation Metrics. Following previous works [Shi *et al.*, 2024; Zhang *et al.*, 2024b; Duarte *et al.*, 2024; Carlini *et al.*, 2022] we use the Area Under the ROC curve (AUC) and True Positive Rate at low False Positive Rate (TPR@5%FPR) as our evaluation metrics. A higher AUC reflects better performance with higher TPR and lower FPR across thresholds.

Baselines. We compare our framework with the following strong baselines. For the grey-box methods, eight methods are selected: **PPL**, **Lowercase**, **Zlib** [Carlini *et al.*, 2021], **Neighbor** [Mattern *et al.*, 2023], **Min-K% Prob** [Shi *et al.*, 2024], **Min-K%++ Prob** [Zhang *et al.*, 2024a], **DC-PDD** [Zhang *et al.*, 2024b], and **FeatureAgg** [Maini *et al.*, 2024]. Two methods are selected for the black-box methods: **DE-COP** [Duarte *et al.*, 2024] and **Name-Cloze** [Chang *et al.*, 2023]. The details are described in Appendix A.

Implementation Details. In our work, all experiments are implemented on a workstation with five NVIDIA Tesla V100 32G GPUs, and Ubuntu22.04.4. For each text to be detected, three suffixes were generated using the target LLM, with the maximum suffix length set to 512 tokens. The parameter γ is set to 10 for obtaining the perturbed text r. The p-value significance threshold is chosen from $\{0.001, 0.01, 0.05, 0.1\}$ to select the critical perturbation calibration features. We randomly sample approximately 50 ground-truth samples per dataset to train the prototype-based classifier, with the remaining samples serving as the texts to be detected. It should

Type	Method	WikiMIA			BookTection			arXivTection		
		Pythia-6.9B	LLaMA-13B	NeoX-20B	Mistral-7B	LLaMA2-7B	ChatGPT	Mistral-7B	LLaMA2-13B	Claude 2.1
Grey-box	PPL	0.635	0.666	0.689	0.698	0.701	✗	0.684	0.658	✗
	Lowercase	0.600	0.602	0.666	0.675	0.697	✗	0.547	0.475	✗
	Zlib	0.645	0.678	0.700	0.539	0.541	✗	0.586	0.566	✗
	Neighbor	0.649	0.654	0.690	0.614	0.637	✗	0.525	0.528	✗
	Min-K% Prob	0.663	0.688	0.736	0.698	0.701	✗	0.749	0.732	✗
	Min-K%++ Prob	0.691	0.847	0.755	0.593	0.610	✗	0.695	0.708	✗
	DC-PDD	0.698	0.697	0.766	–	–	✗	–	–	✗
	FeartureAgg	0.651	0.645	0.728	0.696	0.702	✗	0.774	0.751	✗
Black-box	Name-cloze	0.557	0.534	0.525	0.538	0.538	0.544	0.540	0.527	0.609
	DE-COP	0.512	0.512	0.531	0.674	0.616	0.720	0.551	0.506	0.583
	VeilProbe	**0.913**	**0.902**	**0.916**	**0.960**	**0.963**	**0.947**	**0.797**	**0.837**	**0.904**

Table 2: AUC scores on WikiMIA, BookTection-128, and arXivTection with corresponding target LLMs.

Type	Method	WikiMIA			BookTection			arXivTection		
		Pythia-6.9B	LLaMA-13B	NeoX-20B	Mistral-7B	LLaMA2-7B	ChatGPT	Mistral-7B	LLaMA2-13B	Claude 2.1
Grey-box	PPL	0.140	0.156	0.170	0.214	0.220	✗	0.095	0.083	✗
	Lowercase	0.117	0.123	0.152	0.209	0.243	✗	0.087	0.058	✗
	Zlib	0.178	0.143	0.196	0.134	0.150	✗	0.067	0.062	✗
	Neighbor	0.045	0.116	0.170	0.132	0.144	✗	0.045	0.049	✗
	Min-K% Prob	0.183	0.187	0.233	0.214	0.220	✗	0.197	0.166	✗
	Min-K%++ Prob	0.211	0.369	0.214	0.118	0.148	✗	0.143	0.162	✗
	DC-PDD	0.245	0.230	0.317	–	–	✗	–	–	✗
	FeartureAgg	0.166	0.120	0.229	0.238	0.218	✗	0.257	0.241	✗
Black-box	Name-cloze	0.093	0.070	0.079	0.127	0.114	0.148	0.124	0.096	0.090
	DE-COP	0.061	0.064	0.092	0.208	0.173	0.385	0.067	0.018	0.079
	VeilProbe	**0.625**	**0.554**	**0.625**	**0.807**	**0.845**	**0.723**	**0.367**	**0.411**	**0.581**

Table 3: TPR@5%FPR scores on WikiMIA, BookTection-128, and arXivTection with corresponding target LLMs.

Document	Metric	Mistral-7B	LLaMA2-13B	Claude 2.1	Avg.
165 books	AUC	0.996	0.998	0.988	0.994
	TPR@5%FPR	0.991	1.000	0.943	0.978
	Metric	**Mistral-7B**	**LLaMA2-7B**	**ChatGPT**	**Avg.**
50 papers	AUC	0.978	0.994	0.994	0.989
	TPR@5%FPR	0.920	0.960	0.960	0.947

Table 4: The detection performance of 165 books and 50 papers.

Metric	Approaches	WikiMIA	BookTection
AUC	w/o calibration feature	0.929	0.912
	w/o critical selection	0.935	0.908
	w/o key perturbation	0.931	0.914
	w/o noise reduction	0.929	0.915
	Cosine distance	0.887	0.904
	Manhattan distance	0.934	0.917
	Full	**0.937**	**0.919**
TPR@5%FPR	w/o calibration feature	0.668	0.622
	w/o critical selection	0.684	0.625
	w/o key perturbation	0.703	0.637
	w/o noise reduction	0.719	0.631
	Cosine distance	0.476	0.649
	Manhattan distance	0.668	**0.653**
	Full	**0.731**	0.644

Table 5: Ablation studies on the WikiMIA with Pythia-6.9B and on the BookTection with ChatGPT as the corresponding target LLMs.

be noted that our evaluations are carried out at both sentence and document levels, and the document-level results are derived from the sentence-level ones. Specifically, we compute the average of the sentence-level detection scores within each document to obtain the document-level results.

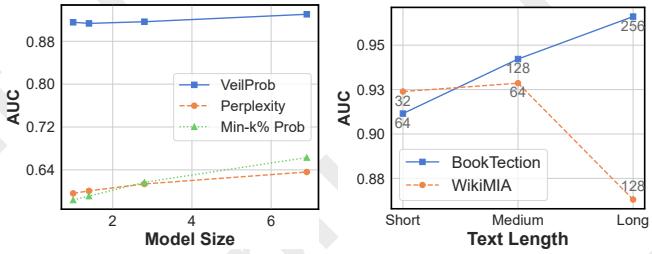
5.2 Main Results

Table 2 and Table 3 show the comparison methods' AUC and TPR@5%FPR scores, respectively. We can observe that (1) VeilProbe significantly outperforms all the other methods, achieving an impressive improvement of over 25.1% in AUC and nearly doubling the TPR@5%FPR scores; (2) Compared to grey-box detection methods, VeilProbe can be implemented on ChatGPT and Claude because it does not rely on hidden information; (3) VeilProbe significantly improves text detection performance compared to existing black-box methods. One possible reason is that these methods often rely on complicated instructions, making some LLMs struggle to comprehend the task.

Additionally, we evaluate document-level detection on 165 bestselling books and 50 arXiv papers from the BookTection and arXivTection datasets by aggregating sequences' detection scores. As shown in Table 4, VeilProbe achieves an average AUC of 99.2% and an average TPR@5%FPR of 96.3% when detecting documents. The detailed results are provided in the Appendix C.4. The document-level results consistently outperform the sentence-level detection, which is in line with previous works [Puerto *et al.*, 2024].

5.3 Ablation Studies

To further assess the importance of each module in **VeilProbe**, we perform ablation studies on WikiMIA-64 and BookTection-64. For the membership feature inference stage, we analyze the contributions of (i) perturbation calibration feature extraction, (ii) critical feature selection with significance tests, and (iii) key token perturbation. For the prototype-based classifier, we evaluate (i) the impact of the IB module and (ii) the impact of different distance metrics



(a) AUC w.r.t Model Size. (b) AUC w.r.t Text Length.

Figure 4: VeilProbe's performance w.r.t different general factors.

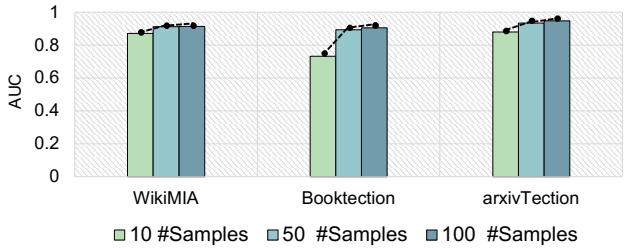


Figure 5: VeilProbe's performance w.r.t #Ground-truth samples.

for prototype computation.

Table 5 shows that all ablation variants demonstrate lower performance than the full framework. The perturbation calibration features are essential, particularly in the TPR@5%FPR score. The critical feature selection can improve performance by reducing noise induced by perturbation operations. Key token perturbations are more effective than random perturbations. The IB module can eliminate redundant information regarding membership status to enhance performance. The most suitable distance function for prototype computing in our task is the Squared Euclidean distance.

5.4 Discussion and Analysis

In this subsection, we analyze five main factors that may affect the detection performance. Among them, the former two are general factors, while the latter three are framework-specific factors.

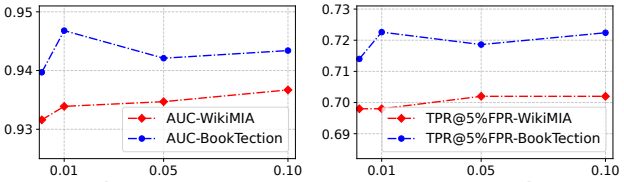
Size of target LLMs. To investigate the impact of different target LLM sizes, we analyze Pythia models of sizes 1B, 1.4B, 2.8B, and 6.9B on WikiMIA. Figure 4a shows that AUC scores improve with increasing model sizes, aligning with previous studies [Shi *et al.*, 2024; Duarte *et al.*, 2024]. Additionally, our proposal consistently outperforms baselines.

Length of texts. We evaluate performance with respect to different text lengths (Figure 4b). For BookTection, the detection performance increases with longer texts. The possible reason is that longer texts can contain more distinguishable features. For WikiMIA, the AUC score decreases at the length of 128, potentially due to the limited number of texts of this length (only 250 samples), much smaller in number than the samples of other lengths, as shown in Table 1.

Number of ground-truth samples. We further explore the impact of the number of ground-truth samples. Figure 5 shows that AUC scores improve as the number of samples increases. Nevertheless, even with only 10 ground-truth samples, our method is still capable of achieving an AUC detection rate of approximately 80-90% across three datasets.

Dataset	Metric	Full	Full-LT	LL	LLT	Avg.
WikiMIA	AUC	**0.937**	0.928	0.924	0.916	0.926
	TPR@5%FPR	**0.731**	0.612	0.573	0.569	0.621
BookTection	AUC	**0.947**	0.937	0.939	0.928	0.938
	TPR@5%FPR	**0.723**	0.702	0.689	0.663	0.694
arXivTection	AUC	0.890	0.880	**0.904**	0.890	0.891
	TPR@5%FPR	**0.569**	0.502	0.581	0.521	0.543

Table 6: VeilProbe's performance w.r.t the volume of text-to-suffix mapping features.



(a) AUC. (b) TPR@5%FPR.

Figure 6: VeilProbe's performance w.r.t p-value.

Volume of text-to-suffix mapping features. Table 6 shows the impact of different volumes of mapping features on the detection performance. We use four extraction methods: (i) `Full`, which extracts the hidden states from all layers of the mapping model; (ii) `Full-LT`, extracting the last token's hidden states from all layers; (iii) `LL`, extracting the hidden states from the last layer of both encoder and decoder; and (iv) `LLT`, extracting the last token's hidden states from the last layer of both encoder and decoder. Despite a general slight decline in performance metrics as the quantity of mapping features is decreased, our approach stays effective at a high detection performance.

Significance threshold for critical perturbation calibration feature selection. Figure 6 shows that the optimal p-value varies across different text domains. A p-value of 0.1 allows tolerance, 0.05 is standard, and 0.001 is stricter. When the p-values are large, there is a risk of underrepresenting crucial features. Conversely, when they are small, it may lead to the introduction of misleading features that impede detection.

6 Conclusion

In this paper, we propose VeilProbe, the first automatic pre-training text detection framework for black-box LLMs. We propose a novel sequence-to-sequence mapping model that can effectively capture the Text-to-Suffix mapping pattern and infer latent membership features. Given the scarcity of ground-truth labeled pre-training and non-training samples, we devise a prototype-based classifier to identify pre-training texts and mitigate the overfitting issue. Comprehensive experiments on three widely used datasets demonstrate that our framework significantly outperforms existing baselines. Notably, VeilProbe operates automatically, without any human intervention throughout the entire detection process.

Acknowledgments

This work was supported by the Natural Science Foundation of China (No. 62372057).

References

[Abbasi-Yadkori *et al.*, 2024] Yasin Abbasi-Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. To believe or not to believe your LLM: Iterativeprompting for estimating epistemic uncertainty. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[Biderman *et al.*, 2023] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023.

[Black *et al.*, 2022] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model, 2022.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, and .etc Subbiah. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

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

[Carlini *et al.*, 2022] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles, 2022.

[Chang *et al.*, 2023] Kent K. Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. Speak, memory: An archaeology of books known to chatgpt/gpt-4, 2023.

[Dong *et al.*, 2024] Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or memorization: Data contamination and trustworthy evaluation for large language models, 2024.

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

[Duarte *et al.*, 2024] André V. Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. De-cop: Detecting copyrighted content in language models training data, 2024.

[Golchin and Surdeanu, 2024] Shahriar Golchin and Mihai Surdeanu. Time travel in llms: Tracing data contamination in large language models, 2024.

[Grynbaum and Mac, 2023] Michael M. Grynbaum and Richard Mac. The times sues openai and microsoft over a.i. use of copyrighted work. *The New York Times*, 2023.

[Hooker *et al.*, 2019] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks, 2019.

[Jiang *et al.*, 2023] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[Karamolegkou *et al.*, 2023] Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models, 2023.

[Kim *et al.*, 2023] Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. Propile: Probing privacy leakage in large language models, 2023.

[Li *et al.*, 2024] Xuhong Li, Jiamin Chen, Yekun Chai, and Haoyi Xiong. Gilot: Interpreting generative language models via optimal transport. In *Forty-first International Conference on Machine Learning*, 2024.

[Liu *et al.*, 2024] Han Liu, Yuhao Wu, Zhiyuan Yu, and Ning Zhang. Please tell me more: Privacy impact of explainability through the lens of membership inference attack. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 4791–4809, 2024.

[Lukas *et al.*, 2023] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models, 2023.

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

[Mattern *et al.*, 2023] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics.

[Mireshghallah *et al.*,] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks.

[Oren *et al.*, 2023] Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto.

Proving test set contamination in black box language models, 2023.

[Puerto *et al.*, 2024] Haritz Puerto, Martin Gubri, Sangdoo Yun, and Seong Joon Oh. Scaling up membership inference: When and how attacks succeed on large language models, 2024.

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

[Shokri *et al.*, 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017.

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

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[Tishby and Zaslavsky, 2015] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle, 2015.

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

[Touvron *et al.*, 2023b] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[Vaswani *et al.*, 2023] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[Wang *et al.*, 2024] Jeffrey G. Wang, Jason Wang, Marvin Li, and Seth Neel. Pandora's white-box: Precise training data detection and extraction in large language models, 2024.

[WELCH, 1947] B. L. WELCH. The generalization of 'student's' problem when several different population varlances are involved. *Biometrika*, 34(1-2):28–35, 01 1947.

[Ye *et al.*, 2022] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, Nov 2022.

[Ye *et al.*, 2024] Wentao Ye, Jiaqi Hu, Liyao Li, Haobo Wang, Gang Chen, and Junbo Zhao. Data contamination calibration for black-box llms, 2024.

[Yu *et al.*, 2023] Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. Bag of tricks for training data extraction from language models, 2023.

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

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

[Zhu *et al.*, 2015] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, 2015.