Identifying Pre-training Data in LLMs: A Neuron Activation-Based Detection Framework

Hongyi Tang*, **Zhihao Zhu***, **Yi Yang**
The Hong Kong University of Science and Technology
{hongyitang,zhihaozhu,imyiyang}@ust.hk

Abstract

The performance of large language models (LLMs) is closely tied to their training data, which can include copyrighted material or private information, raising legal and ethical concerns. Additionally, LLMs face criticism for dataset contamination and internalizing biases. To address these issues, the Pre-Training Data Detection (PDD) task was proposed to identify if specific data was included in an LLM's pre-training corpus. However, existing PDD methods often rely on superficial features like prediction confidence and loss, resulting in mediocre performance. To improve this, we introduce NA-PDD, a novel algorithm analyzing differential neuron activation patterns between training and non-training data in LLMs. This is based on the observation that these data types activate different neurons during LLM inference. We also introduce CCNewsPDD, a temporally unbiased benchmark employing rigorous data transformations to ensure consistent time distributions between training and non-training data. Our experiments demonstrate that NA-PDD significantly outperforms existing methods across three benchmarks and multiple LLMs. Our code is available at https://github.com/tanghongyi0406/CCNewsPDD

1 Introduction

The effectiveness of large language models (LLMs) hinges significantly on their training corpus (Kaplan et al., 2020; Gao et al., 2020). However, these pre-training corpora may contain copyrighted material (Chang et al., 2023; Mozes et al., 2023) or private user information (Yao et al., 2024b; Liu et al., 2021), raising substantial concerns about compliance and privacy. For example, The New York Times recently filed a lawsuit against OpenAI, alleging illegal use of its articles as training data for ChatGPT [1]. Furthermore, LLMs can inadvertently acquire undesirable knowledge from their training data, such as biased (Ferrara, 2023; Kotek et al., 2023) or harmful content (Deshpande et al., 2023; Gehman et al., 2020), compromising the trustworthiness of the language model. Precise knowledge of the learned data is therefore crucial. However, determining whether a model has incorporated specific data remains challenging. This leads to a critical question: *given an LLM and a text sample, how can we determine if this text was part of the LLM's pre-training?* This is the pre-training data detection (PDD) problem.

Existing PDD algorithms suffer from two primary limitations: 1) **Superficial Information Reliance**: Most algorithms focus on surface-level features of LLMs (Carlini et al., 2023; Zhang et al., 2024b). For instance, Loss Attack (Yeom et al., 2018) uses the LLM's prediction loss on a given text, while Min-K% Prob (Shi et al., 2023a) uses predictive probabilities of tokens. This approach limits detection effectiveness, resulting in insufficient performance and high false positive rates, rendering them unsuitable for applications such as copyright verification (Duan et al., 2024a; Zhang et al., 2024a). 2) **Benchmark Time Drift**: Due to the confidentiality of LLM training data (Achiam et al., 2023; Bai et al., 2023), researchers often use release dates to infer training data, comparing it to publicly available datasets like Wikipedia. For example, pre-2023 data might be considered training data for a 2023 LLaMA model, while post-2023 data is viewed as non-training data (Shi et al., 2023a). This temporal bias complicates the accurate evaluation of PDD methods intended to identify training corpora.

To address the first limitation, we introduce **NA-PDD**, a novel PDD algorithm that utilizes neuronal activation patterns within LLMs. Our method stems from the observation that training text activates different neurons within an LLM compared to non-training text. NA-PDD is particularly suitable

*Equal contribution

[1] https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 18739–18752
November 4-9, 2025

for copyright verification in open-source LLMs or internal audits. It utilizes a small set of reference corpora to record neuronal activation for both training and non-training data. Neurons predominantly activated by training data are labeled as "member" neurons, while those activated by non-training data are labeled as "non-member" neurons. We then design a straightforward detection algorithm to determine whether a given sample x was part of the LLM's pre-training corpus. During model inference with input x, we record the activation states of neurons across different layers and provide PDD predictions based on the relative prominence of member neurons in these layers.

To address the second limitation, we introduce **CCNewsPDD**, a time-drift-free PDD benchmark based on the CCNews dataset. This benchmark ensures temporal alignment between training and non-training data. To make sure that non-training data was not used in pre-training, we apply transformations such as back translation, masking, and LLM rewriting to the original non-training data. These transformations introduce meaningful variations while maintaining a rigorous definition of non-training data.

Our contributions are as follows:

- We introduce NA-PDD, a novel PDD algorithm leveraging neuronal activation patterns within LLMs. NA-PDD analyzes the differential activation between trained and non-trained samples during inference to construct an effective PDD algorithm.
- We introduce CCNewsPDD, a time-drift-free PDD benchmark. Using data transformation methods, CCNewsPDD ensures no temporal distribution differences between training and non-training data while maintaining semantic and lexical coherence.
- We evaluate NA-PDD against nine representative PDD methods on CCNewsPDD and two public benchmarks. Our results demonstrate substantial improvements. For example, on OPT-6.7B with CCNewsPDD, NA-PDD outperforms Smaller Model by 26.5% AUC points (increasing from 67.1% to 93.6%).

2 Related Work

Membership Inference Attacks (MIA). Membership inference attacks determine whether specific data was used to train a model (Shokri et al., 2017; Hu et al., 2022). Originating in genomics (Homer et al., 2008; Pyrgelis et al., 2017), this field evolved within machine learning through shadow modeling and black-box techniques (Salem et al., 2018; Yeom et al., 2018). MIA research has expanded across computer vision (Choquette-Choo et al., 2021), generative models (Chen et al., 2020), and diffusion systems (Carlini et al., 2023), while defensive strategies like differentially private training emerged in parallel (Abadi et al., 2016; Jia et al., 2019). Recently, MIA has become crucial for Large Language Models, detecting memorized training data (Nasr et al., 2023; Oren et al., 2023) and potential copyright issues (Duarte et al., 2024; Meeus et al., 2024). Our work focuses specifically on the the precise detection of pre-training data.

LLM Pretraining Data Detection. Traditional MIAs predominantly employ black-box approaches, relying solely on model output signals for inference (Yeom et al., 2018; Sablayrolles et al., 2019). In contrast, we adopt a white-box strategy, directly accessing the model's internal states. In NLP, previous research includes likelihood ratio attacks on causal language models by Carlini et al. (2021), neighborhood attacks proposed by Mattern et al. (2023), and membership inference based on outlier word likelihoods and probability distribution features by Shi et al. (2023a), Mireshghallah et al. (2022), and Watson et al. (2021). Unlike these approaches that depend on surface features, we investigate LLM internal neuron activation states, exploring their specific memorization characteristics for pre-training data, thereby achieving enhanced detection performance.

3 Methodology

In this section, we begin by introducing the task of pre-training data detection in large language models. We then provide a detailed description of our method, NA-PDD, a pre-training data detection algorithm that captures the differences in neuronal activation between the pre-training corpus and other data.

3.1 Problem Statement

Pre-training data detection (also known as membership inference) aims to determine whether a large language model (LLM) has utilized specific data points, such as text, during its training phase. Formally, for a given text x and a target LLM $\mathcal{M}$,

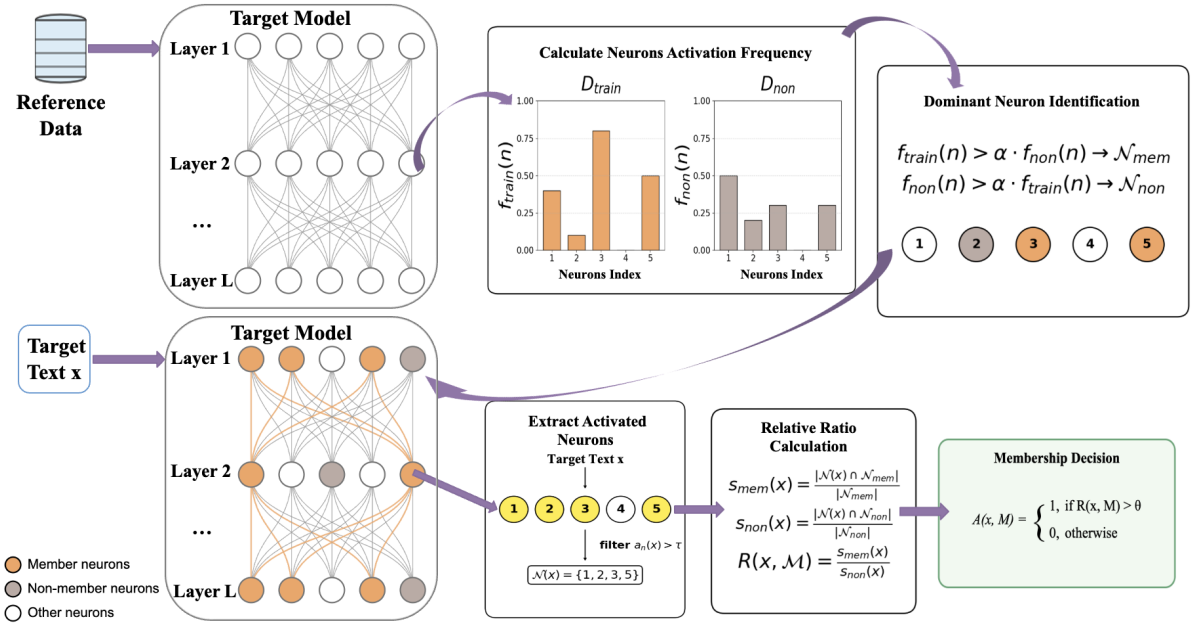

Figure 1: An overview of NA-PDD.

the detection algorithm is structured as a binary classification problem:

$$\mathcal{A}(x, \mathcal{M}) \rightarrow \{0, 1\} \tag{1}$$

where a prediction of **1** indicates that the model $\mathcal{M}$ has utilized the text x, while a prediction of **0** signifies that it has not.

White-box settings. Following previous work, we assume access to the weights and activations of the target model. This framework is applicable in two real-world scenarios: 1) Model owners need to audit their models to prevent dataset contamination (Xu et al., 2024; Magar and Schwartz, 2022) or to assess the effectiveness of machine unlearning (Bourtoule et al., 2021; Yao et al., 2024a); 2) Data owners need to verify whether their data has been incorporated into an open-source model (Mökander et al., 2024; Pan et al., 2020).

3.2 Overview

We now propose NA-PDD, which is outlined in Figure 1. Our design is based on the observation that the language model tends to activate specific neurons when inferring with training data as opposed to other data. NA-PDD consists of the following four steps: (i) *Neuronal Activation Determination*: This phase detects active neurons by analyzing their activation values as the target model $\mathcal{M}$ processes the given text x (Section 3.3). (ii) *Neuronal Identity Discrimination*: By examining the activation status of neurons in response to different types of data (training data and non-training data), we label neurons that are more easily activated by training data as "member neurons," while those that respond more to non-training data are labeled as "non-member neurons" (Section 3.4). (iii) *Neuronal Similarity Calculation*: To ascertain whether a target text x was part of the training data for a large language model, we evaluate the relationship between member neurons and x by examining the neurons activated when x is input into $\mathcal{M}$ (Section 3.5). (iv) *Membership Inference*: By comparing the advantage of member neurons for the target text x across different layers against a predefined threshold, we predict whether the text was part of the pre-training corpus of the target model (Section 3.6). Our method is summarized in Algorithm 1, which is detailed in Appendix A.

3.3 Neuronal Activation Determination

The sparsely activated nature of large language models suggests (Wang et al., 2024; Liu et al., 2024a) that only a few neurons are activated when processing text x. These neurons have a significant influence on the prediction of x. To identify which neurons play a major role in the inference process of the target model on specific training corpora (e.g., certain news texts used for pre-training), we establish a flexible threshold τ to determine the activation state of neurons.

Formally, given input text x to the target model $\mathcal{M}$, which consists of neurons $\mathcal{N}$, the activation

state of neuron $n \in \mathcal{N}$ is defined as follows:

$$I(x,n) = \begin{cases} 1, & \text{if } a_n(x) > \tau, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

where $a_n(x)$ denotes the output value of neuron n for text x. Neurons with an output value greater than the threshold τ are considered to be "active".

3.4 Neuronal Identity Discrimination

We introduce reference corpora to identify neurons that tend to be activated when processing training or non-training data. The reference data can originate from the model owner's training database (Liu et al., 2024b). Specifically, for a set of collected training data $\mathcal{D}_{train}$, we record the activation frequency of all neurons n in the target model $\mathcal{M}$:

$$f_{train}(n) = \frac{1}{|\mathcal{D}_{train}|} \sum_{x \in \mathcal{D}_{train}} I(x,n) \quad (3)$$

where $f_{train}(n)$ represents the frequency at which neurons n are activated for the training data. The frequency of neuron activation for non-training data is calculated in a similar manner:

$$f_{non}(n) = \frac{1}{|\mathcal{D}_{non}|} \sum_{x \in \mathcal{D}_{non}} I(x,n) \quad (4)$$

The frequency at which neurons are activated can help identify which neurons play a more significant role in the prediction process. However, relying solely on these activation frequencies does not effectively distinguish between the activation of neurons for training texts and non-training texts. For some low-level lexical neurons, activation occurs frequently for both types of texts as these neurons extract common features. Consequently, focusing on these neurons is insufficient for confirming whether the model is utilizing specific data during training. Therefore, we assess the relative dominance of neuron activation frequency to determine whether a neuron is dominant in the training data, using the following formula:

$$\mathcal{N}_{mem} = \{n \in \mathcal{N} \mid f_{train}(n) > \alpha \cdot f_{non}(n)\} \quad (5)$$

Neurons that are more likely to activate in response to training texts, compared to non-training texts, are referred to as "member" neurons. Similarly, the computation process for identifying "non-member" neurons is as follows:

$$\mathcal{N}_{non} = \{n \in \mathcal{N} \mid f_{non}(n) > \alpha \cdot f_{train}(n)\} \quad (6)$$

where $\alpha > 1$ serves as the dominance threshold, controlling the stringency for classifying neurons as "member" or "non-member" neurons.

3.5 Neuronal Similarity Calculation

The powerful learning capabilities of LLMs often lead to noticeable memory effects on training data. This characteristic is a key reason why pre-training data detection algorithms are effective (Carlini et al., 2021; Tirumala et al., 2022). We demonstrate that, beyond mere superficial features or hidden units, the activation patterns of neurons can more accurately reflect the memory phenomena within language models. Specifically, the model's retention and assimilation of the training corpus enable it to pinpoint specific neurons that enhance and accelerate the reasoning process for similar texts. Building on this intuition, if text x was used in training the target model $\mathcal{M}$, the neurons activated during the inference process will show a higher similarity to those activated by other training data. We measure this similarity through the coincidence rate of activated neurons.

$$s_{mem}(x) = \frac{|\mathcal{N}(x) \cap \mathcal{N}_{mem}|}{|\mathcal{N}_{mem}|} \quad (7)$$

where $\mathcal{N}(x)$ represents the set of neurons activated for text x according to Eq.2. Similarly, the coincidence rate of neurons activated by the text x with non-member neurons is calculated as follows:

$$s_{non}(x) = \frac{|\mathcal{N}(x) \cap \mathcal{N}_{non}|}{|\mathcal{N}_{non}|} \quad (8)$$

3.6 Membership Inference

Different layers of large language models often undertake diverse roles during the inference process (Zhao et al., 2024) and exhibit varying performance on pre-training data detection task (Liu et al., 2024c). Therefore, we propose a scoring mechanism to evaluate the capability of different neuronal layers to differentiate between training and non-training data as follows:

$$S_\ell = |\mathcal{N}^\ell_{mem}| - |\mathcal{N}^\ell_{non}| \quad (9)$$

where $\mathcal{N}^\ell_{mem}$ and $\mathcal{N}^\ell_{non}$ represent the member and non-member neurons of the ℓ-layer, respectively. The higher the score, the greater the imbalance between member and non-member neurons within that layer. Based on the difference scores S_ℓ, we select the K most discriminative layers $\mathcal{L}_{dis}$ and compute the average activation neuron similarity from the selected layers:

$$\bar{s}_{mem}(x) = \frac{1}{|\mathcal{L}_{dis}|} \sum_{\ell \in \mathcal{L}_{dis}} s^\ell_{mem}(x) \quad (10)$$

$$\bar{s}_{non}(x) = \frac{1}{|\mathcal{L}_{dis}|} \sum_{\ell \in \mathcal{L}_{dis}} s^{\ell}_{non}(x) \tag{11}$$

The member advantage for text x is defined by the ratio of $\bar{s}_{\text{mem}}(x)$ to $\bar{s}_{\text{non}}(x)$. A higher ratio suggests a greater likelihood that text x was part of the training data for the target model $\mathcal{M}$.

$$R(x, \mathcal{M}) = \frac{\bar{s}_{mem}(x)}{\bar{s}_{non}(x)} \tag{12}$$

After calculating the member advantage $R(x, \mathcal{M})$ for text x, we predict whether x was included in model $\mathcal{M}$'s pre-training data by applying a predefined threshold θ to $R(x, \mathcal{M})$:

$$\mathcal{A}(x, \mathcal{M}) = \begin{cases} 1, & \text{if } R(x, \mathcal{M}) > \theta, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

4 Data Construction

As proprietary information, the training logs of LLMs are generally not publicly accessible. To evaluate the performance of pre-training data detection (PDD) algorithms on LLMs, researchers typically rely on commonly used benchmark datasets (e.g., Wikipedia) as proxies for the models' training data. Non-training data, in contrast, is curated based on the release timelines of the target LLMs (Shi et al., 2023b; Liu et al., 2024c). For example, the Pythia model (2023 release) (Biderman et al., 2023) cannot have been trained on Wikipedia articles published after 2024. However, this temporal partitioning of training versus non-training data introduces a critical time drift confounder. This methodological limitation raises questions about whether the observed efficacy of PDD algorithms genuinely reflects their discriminative capability or merely artifacts of the dataset construction methodology (e.g., temporal distributional shifts rather than true memorization signals) (Maini and Suri, 2025; Duan et al., 2024b).

To address this limitation, we propose **CC-NewsPDD**, a carefully designed benchmark for PDD evaluation that effectively eliminates temporal confounding effects. For member data, we sample from Pile-CC dataset [2], which is known to be included in the pre-training corpora of major LLMs such as Pythia (Biderman et al., 2023) and OPT (Zhang et al., 2022). For non-member data, we use the CC-News dataset prepared by news-please (Hamborg et al., 2017) and restrict the data to articles published in 2017. This specific year predates the primary data collection windows for the target LLMs, allowing us to create a benchmark that effectively eliminates temporal confounding effects.

This non-member data is not part of the target LLMs' known pre-training corpora. Given the vast size of the full pre-training corpora, it is highly probable that our specific subset was excluded from their training data. However, to elevate this high probability to a strict guarantee that the texts are genuinely unseen by the target models while maintaining their authentic linguistic properties, we apply a series of principled data transformations. We implement three such transformations on the non-member data, carefully designed to maintain their authentic linguistic properties. These transformations are meticulously designed to preserve the original data distribution while introducing meaningful variations, thereby creating a rigorous testbed for evaluating the PDD algorithm. From this process, we derive three complementary datasets that together offer a comprehensive assessment framework free from temporal bias:

CCNewsPDD(trans): This dataset implements sequential translation from English to French and back to English using MarianMT models (Tiedemann et al., 2024). This approach generates semantically preserved non-training data with diverse syntactic variations.

CCNewsPDD(mask): This dataset randomly masks 15% of tokens in the original text and uses BERT (Devlin et al., 2019) to predict contextually appropriate substitutions. Unlike back-translation, this strategy focuses on localized perturbations while maintaining the global text structure.

CCNewsPDD(prompt): This dataset generates non-training data through explicit instruction prompting. By directing the GPT-4o model (Lewis et al., 2020) to reformulate the original texts, we achieve comprehensive discourse-level rephrasings while preserving the core semantic content in the generated non-training data.

5 Experimental Settings

Benchmarks and Models. We evaluate NA-PDD's performance across three benchmark datasets (Table 1). For ArxivMIA (Liu et al., 2024c), comprising arXiv academic paper abstracts, we follow the original study's experimental setting by testing TinyLLaMA-1.1B (Zhang et al., 2024c)

[2] https://huggingface.co/datasets/monology/pile-uncopyrighted

Benchmark	Data source	Text length	#Examples	Applicable models
WikiMIA (Shi et al., 2023a)	Wikipedia	32	774	Pythia-2.8B, OPT-6.7B
ArxivMIA (Liu et al., 2024c)	Arxiv	143.1	2,000	TinyLLaMA-1.1B, OpenLLaMA-13B
CCNewsPDD (**Ours**)	CC-news	309.1	1,200	Pythia-2.8B, OPT-6.7B

Table 1: Benchmark summary statistics: Each benchmark has an equal split of training and non-training examples. "Text Length" refers to the average number of tokens in each text example of the benchmark. "#Examples" denotes the total number of text examples in the benchmark.

and OpenLLaMA-13B (Geng and Liu, 2023). For WikiMIA (Shi et al., 2023a) and our CCNewsPDD benchmark, we select Pythia-2.8B (Biderman et al., 2023) and OPT-6.7B (Zhang et al., 2022) as our evaluation models, as their pretraining corpora are known to include Wikipedia dumps and the CC-News dataset, respectively.

Baselines. We evaluate NA-PDD against nine state-of-the-art pretraining data detection (PDD) methods, which include both reference-free and reference-based approaches. The reference-free methods are: (i) *Loss Attack* (Yeom et al., 2018), which utilizes the target model's loss values; (ii) *Neighbor Attack* (Mattern et al., 2023), which compares loss values between target samples and synthetically generated neighbor texts; (iii) *Min-K% Prob* (Shi et al., 2023a), which analyzes the average log-likelihood of the least probable tokens; (iv) *Min-K%++ Prob* (Zhang et al., 2024b), which extends this approach with vocabulary-normalized probabilities; and (v) *DC-PDD* (Zhang et al., 2024d), which proposes a divergence-based calibration Method for pretraining data detection.

The reference-based methods, which employ a reference (proxy) model for detection calibration, include: (vi) *Zlib* (Carlini et al., 2021), which computes the ratio between an example's perplexity and its zlib entropy; (vii) *Lowercase* (Carlini et al., 2021), which compares perplexity between original and lowercased text; (viii) *Small Ref* (Carlini et al., 2021), which uses perplexity ratios between target and reference models; and (ix) *Probe Attack* (Liu et al., 2024c), which examines internal model activations through probing techniques.

Evaluation Metrics. Following prior work (Shi et al., 2023a; Zhang et al., 2024d), we use the Area Under the ROC Curve (AUC) to assess detection performance. AUC provides a threshold-independent measure of pretraining data detection performance and is robust to class imbalance.

Implementation Details. We introduce the details of NA-PDD through three core components: (i) *Neuronal Activation Determination*, where we attach hook functions to all feed-forward network (FFN) layers in the Transformer architecture to record post-activation outputs of neurons; (ii) *Neuronal Identity Discrimination*, which utilizes 100 training and 100 non-training samples to classify neurons as either member or non-member neurons; and (iii) *Hyperparameter Selection*, where extensive ablation studies determine the final configuration: activation threshold $\tau = 1.0$ (validated range [0,2]), dominance threshold $\alpha = 1.5$ (tested range [1.2,2.0]), and the number of used discriminative layers $K = 10$ (tested range [1, 32]). For more details on our baselines, please refer to Appendix C.

6 Experimental Results

We report the results of the experiment to address the following questions: **Q1**: What is the performance of NA-PDD across different datasets and language models? **Q2**: How do the size of the model and the amount of reference data impact the performance of the PDD algorithm? **Q3**: Sensitivity analysis: How does NA-PDD perform at different activation threshold τ, dominance threshold α, and numbers of used discriminative layers K?

6.1 Main Results

Table 2 presents the performance comparison of the PDD algorithm across different datasets, leading to four key findings:

Leading Performance of NA-PDD: Our method achieves the best performance on all datasets. Notably, in detecting whether the CC-NewsPDD(prompt) dataset was used in OPT model training, NA-PDD significantly increases the AUROC value from 67.1% to 93.6% compared to the second best algorithm. This confirms neuronal activation's utility in PDD, as NA-PDD reliably distinguishes training data through activation pattern analysis.

Limited Advantages of Reference Model-Based

Method	ArxivMIA		WikiMIA		CCNewsPDD(trans)		CCNewsPDD(mask)		CCNewsPDD(prompt)	
	TinyL.	OpenL.	Pythia	OPT	Pythia	OPT	Pythia	OPT	Pythia	OPT
Reference-free										
Loss Attack	45.1	49.1	65.8	64.5	55.2	52.6	75.7	68.5	48.7	46.7
Neighbor Attack	55.9	56.2	66.0	65.6	63.8	61.7	78.1	77.0	54.3	52.7
Min-K% Prob	45.5	49.2	64.2	64.9	60.6	62.4	76.8	72.7	52.4	50.9
Min-K++% Prob	47.8	52.0	64.9	67.3	71.0	76.6	68.7	74.2	58.5	65.4
DC-PDD	46.2	48.7	63.3	65.4	59.4	60.7	76.5	72.1	49.8	51.2
Reference-based										
Zlib Compression	43.0	43.8	66.5	65.1	69.9	68.0	79.4	75.6	62.0	60.4
Lowercased Text	45.8	48.8	60.9	61.5	57.5	55.6	64.7	62.4	54.5	52.1
Smaller Model	—	56.7	58.1	63.4	54.0	66.2	48.1	55.4	52.9	67.1
Probe Attack	53.2	59.0	69.8	68.1	65.3	74.0	81.3	76.4	70.2	67.0
Our Method										
NA-PDD	**57.2**	**59.3**	**75.8**	**71.6**	**92.4**	**92.1**	**98.8**	**95.2**	**93.2**	**93.6**

Table 2: AUC scores (in %) for various methods across ArxivMIA, WikiMIA, and CCNewsPDD datasets. Best performances in each column are highlighted in bold.

Approaches: While reference-based methods necessitate extra computational resources for prediction calibration, their performance gains are not significant. For example, when evaluating OPT in CCNewsPDD(trans), the leading reference-based method, Probe Attack (68.8% AUROC), was outperformed by the reference-free method, Min-K++% Prob (76.6% AUROC). This indicates that reference-free methods retain potential, as evidenced by our NA-PDD approach, which achieves outstanding detection results solely through neuronal activation analysis.

Detection Difficulty Variations Between Datasets: The ArxivMIA dataset posed significant detection challenges, with the highest AUC (57.2%) among all methods being considerably lower than those for WikiMIA (75.8%) and CCNewsPDD (98.8%). We speculate that this may be due to ArxivMIA containing texts with technical terminology and mathematical formulations, which makes it more challenging for models to memorize training texts and thus complicates the PDD task.

Detection Difficulty Variations Between Models: On the CCNewsPDD(mask) dataset, NA-PDD achieved better performance with the smaller Pythia-2.8B model (98.8% AUC) compared to the larger OPT-6.7B model (95.2% AUC). This performance gap likely arises from their different training procedures, which may lead to the two models facing different challenges in PDD tasks.

6.2 Impact of Different Factors

In this section, we analyze how model size and the amount of reference data affect the performance of the PDD algorithm. To highlight the effectiveness and robustness of NA-PDD, we compare it with Probe Attack, the second-best algorithm according to our main results.

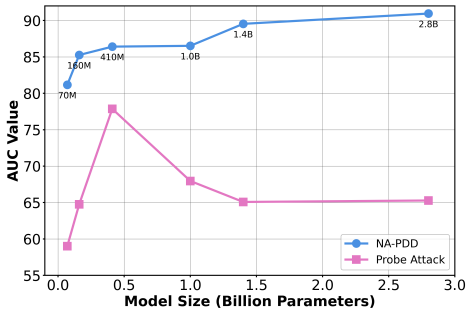

Figure 2: Comparison of AUC Values with Different Model Sizes (best viewed in color).

Model Size. We compare NA-PDD with the Probe Attack baseline across various sizes of Pythia models (ranging from 70M to 2.8B) on the CCNewsPDD(trans) dataset. As shown in Figure 2, our method consistently improves with increasing model size, indicating that larger models develop more distinctive neuron activation patterns for detecting pretraining data. In contrast, the baseline method exhibits unstable behavior: after peaking at 78% AUC with the 410M model, its performance sharply declines and stabilizes around 65% for models larger than 1.4B. This divergence suggests that while traditional probe-based approaches struggle with increased model complexity, our neuron-based detection method effectively utilizes the richer representations in larger models. The widening performance gap highlights the scalability advantage of our approach.

Reference Data Size. We assess the robustness of our method by varying the reference data sizes (200-500 samples) for neuron classification using the Pythia-2.8B model on the CCNewsPDD(trans)

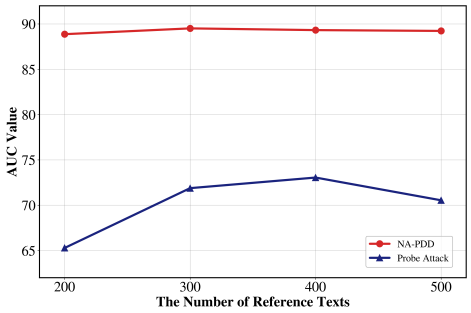

Figure 3: Comparison of AUC Values with Different Training Data Sizes (best viewed in color).

dataset. As illustrated in Figure 3, our method consistently achieves high performance (nearly 90% AUC) across all data sizes, highlighting its data efficiency. In contrast, the baseline Probe Attack is more sensitive to the volume of reference data. Our approach maintains high performance with limited reference data and significantly outperforms all baselines by 16–34%, demonstrating both its data efficiency and robustness.

6.3 Sensitivity Analysis

We further conduct sensitivity analyses on three key hyperparameters: the activation threshold τ, the dominance threshold α, and the numbers of selected discriminative layers K.

Activation Threshold τ. We evaluate the robustness of our membership inference approach with varying activation thresholds $\tau \in [0.0, 2.0]$ across three distinct benchmarks. As illustrated in Figure 4, our method shows remarkable resilience to changes in activation threshold, despite performance differences across datasets. For instance, CCNewsPDD(prompt) and CCNewsPDD(mask) variants consistently reach peak performance with an AUC of approximately 99% across the entire threshold range. In contrast, the CCNewsPDD(trans) variant, despite starting at a lower performance, stabilizes above 90% for $\tau \geq 0.2$.

Dominance Threshold α. We assess the impact of the dominance threshold α on the CCNewsPDD(trans) dataset. As depicted in Figure 5, the AUC remains stable (90.9%–91.25%) across different α values, exhibiting a slight U-shaped trend: starting at 91.05% ($\alpha = 1.2$), dipping to 90.90% ($\alpha = 1.4$), and then peaking at 91.25% ($\alpha = 2.0$). The minimal variation of 0.35% demonstrates the robustness of our method to the selection of α, with $\alpha = 1.5$ recommended as a reliable default.

Numbers of selected discriminative layers K.

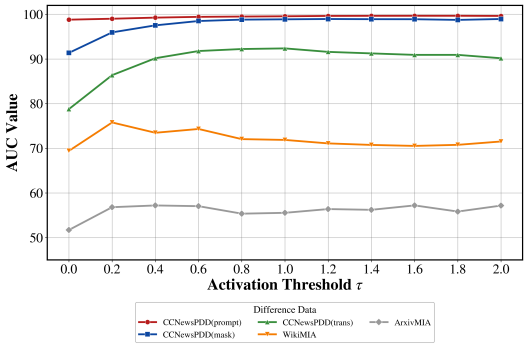

Figure 4: Comparison of AUC Values with Different Activation Thresholds.

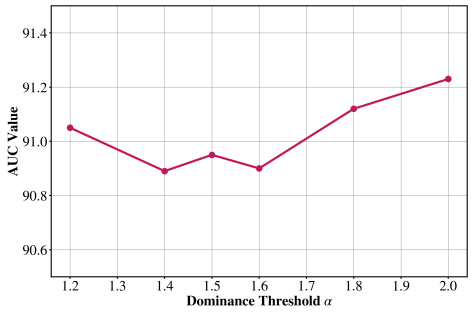

Figure 5: Comparison of AUC Values with Different Dominance Thresholds.

Our evaluation on the Pythia-2.8B model and CCNewsPDD(trans) dataset shows stable detection performance of NA-PDD across varying K values (Fig. 6): AUC increases from 89.6% ($K = 1$) to a peak of 91.2% ($K = 5$), maintaining over 90% for $K \geq 5$ (range: 90.2%–91.2%). The minimal variation highlights our algorithm's robustness, with $K = 5$ providing a good balance between efficiency and performance, while permitting flexible selection of K (from 5 to 32) without significant performance loss.

7 Conclusion

In this paper, we propose NA-PDD, a PDD algorithm that distinguishes between training and non-training data by analyzing neuronal activation

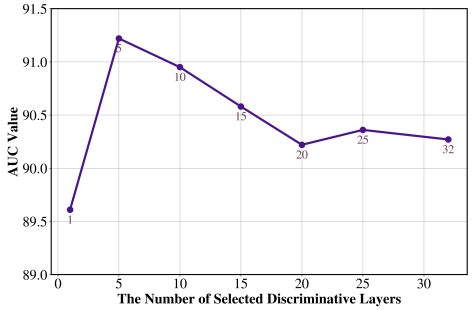

Figure 6: Comparison of AUC Values with Different Numbers of Selected Discriminative Layers.

during LLM inference. Our experimental results show that NA-PDD outperforms baselines across various LLMs and evaluation benchmarks, demonstrating robust performance and insensitivity to hyperparameters. In future work, we aim to explore more complex neuronal activation patterns, such as how activation pathways can further enhance PDD.

Limitations

While NA-PDD shows promising results in detecting pretraining data, it has several limitations. (i) NA-PDD relies on capturing neuron activation patterns for both training and non-training data. Consequently, it is only applicable when the weights of large language models (LLMs) and activation information are accessible. This limitation means it cannot be used with closed-source models or LLMs that only provide a query interface. (ii) NA-PDD requires a portion of a reference corpus to help label member and non-member neurons. Although obtaining these reference corpora is feasible in scenarios like copyright verification and LLM internal audits, and the amount needed is relatively small, NA-PDD is not suitable for situations lacking a reference corpus. (iii) Due to computational resource constraints, NA-PDD has only been evaluated on LLMs with up to 13 billion parameters. However, previous work and our experiments suggest that the PDD algorithm performs better with larger LLMs, making the potential performance of NA-PDD on larger models promising. We plan to explore this in future work.

Acknowledgement

This work is partially supported by a research grant provided by HSBC.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

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

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE.

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

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

Kent K Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to chatgpt/gpt-4. *arXiv preprint arXiv:2305.00118*.

Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 343–362.

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.

Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. *arXiv preprint arXiv:2304.05335*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia

Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024a. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024b. Do membership inference attacks work on large language models? In *First Conference on Language Modeling*.

André V Duarte, Xuandong Zhao, Arlindo L Oliveira, and Lei Li. 2024. De-cop: Detecting copyrighted content in language models training data. *arXiv preprint arXiv:2402.09910*.

Emilio Ferrara. 2023. Should chatgpt be biased? challenges and risks of bias in large language models. *arXiv preprint arXiv:2304.03738*.

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

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama. *URL: https://github.com/openlm-research/open_llama*.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223.

Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. 2008. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167.

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37.

Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 259–274.

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*.

Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in large language models. In *Proceedings of the ACM collective intelligence conference*, pages 12–24.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. 2021. When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

James Liu, Pragaash Ponnusamy, Tianle Cai, Han Guo, Yoon Kim, and Ben Athiwaratkun. 2024a. Training-free activation sparsity in large language models. *arXiv preprint arXiv:2408.14690*.

Yang Liu, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. 2024b. Datasets for large language models: A comprehensive survey. *arXiv preprint arXiv:2402.18041*.

Zhenhua Liu, Tong Zhu, Chuanyuan Tan, Bing Liu, Haonan Lu, and Wenliang Chen. 2024c. Probing language models for pre-training data detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1576–1587.

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.

Pratyush Maini and Anshuman Suri. 2025. Reassessing emnlp 2024's best paper: Does divergence-based calibration for mias hold up? In *The Fourth Blogpost Track at ICLR 2025*.

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*.

Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. 2024. Did the neurons read your book? document-level membership inference for large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 2369–2385.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022. Quantifying privacy risks of masked language

models using membership inference attacks. *arXiv preprint arXiv:2203.03929*.

Jakob Mökander, Jonas Schuett, Hannah Rose Kirk, and Luciano Floridi. 2024. Auditing large language models: a three-layered approach. *AI and Ethics*, 4(4):1085–1115.

Maximilian Mozes, Xuanli He, Bennett Kleinberg, and Lewis D Griffin. 2023. Use of llms for illicit purposes: Threats, prevention measures, and vulnerabilities. *arXiv preprint arXiv:2308.12833*.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

Yonatan Oren, Nicole Meister, Niladri S Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. 2023. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*.

Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. 2020. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1314–1331. IEEE.

Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2017. Knock knock, who's there? membership inference on aggregate location data. *arXiv preprint arXiv:1708.06145*.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR.

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*.

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

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023b. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

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

Jörg Tiedemann, Mikko Aulamo, Daria Bakshandaeva, Michele Boggia, Stig-Arne Grönroos, Tommi Nieminen, Alessandro Raganato, Yves Scherrer, Raúl Vázquez, and Sami Virpioja. 2024. Democratizing neural machine translation with opus-mt. *Language Resources and Evaluation*, 58(2):713–755.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290.

Hongyu Wang, Shuming Ma, Ruiping Wang, and Furu Wei. 2024. Q-sparse: All large language models can be fully sparsely-activated. *arXiv preprint arXiv:2407.10969*.

Lauren Watson, Chuan Guo, Graham Cormode, and Alex Sablayrolles. 2021. On the importance of difficulty calibration in membership inference attacks. *arXiv preprint arXiv:2111.08440*.

Cheng Xu, Shuhao Guan, Derek Greene, M Kechadi, and 1 others. 2024. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024a. Machine unlearning of pre-trained large language models. In *62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024*, pages 8403–8419. Association for Computational Linguistics (ACL).

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024b. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.

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.

Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. 2024a. Membership inference attacks cannot prove that a model was trained on your data. *arXiv preprint arXiv:2409.19798*.

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

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024c. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1

others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024d. Pretraining data detection for large language models: A divergence-based calibration method. *arXiv preprint arXiv:2409.14781*.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

A Algorithm

Algorithm 1 Neuron Activation Pre-training Data Detection (NAPDD)

Require: x; M with neurons $\mathcal{N}$, L layers;
$D_{\text{train}}, D_{\text{non}}$; $\tau, \alpha > 1.5, \theta$; $K \leq L$
Ensure: $A(x, M) \in \{0, 1\}$
Forward x through M; collect $\{a_n(x)\}$.
for all $n \in \mathcal{N}$ **do**
    Extract activation $I(x, n) = 1[a_n(x) > \tau]$, w.r.t. Eq. 2
end for
for all $n \in \mathcal{N}$ **do**
    Compute activation frequency $f_{\text{train}}(n)$ and $f_{\text{non}}(n)$, w.r.t. Eqs. 3, 4
end for
Define dominant neuron $N_{\text{mem}}, N_{\text{non}}$, w.r.t. Eqs. 5, 6
for $\ell = 1$ **to** L **do**
    Compute discriminative score S_ℓ, w.r.t. Eq. 9
end for
$L_{\text{dis}} \leftarrow$ top-K layers by S_ℓ
for all $\ell \in L_{\text{dis}}$ **do**
    Compute neuronal similarity $s^\ell_{\text{mem}}(x)$ and $s^\ell_{\text{non}}(x)$, w.r.t. Eqs. 7, 8
end for
Calculate $\bar{s}_{\text{mem}}, \bar{s}_{\text{non}}$, w.r.t. Eqs. 10, 11
Get ratio $R(x, \mathcal{M})$, w.r.t. Eq. 12
if $R > \theta$ **then**
    $A(x, M) \leftarrow 1$, w.r.t. Eq. 13
else
    $A(x, M) \leftarrow 0$
end if
return $A(x, M)$

B TPR@5% FPR

We evaluate performance under a low false-positive setting, which is critical for practical applications. Table 3 reports the TPR@5%FPR, which means true positive rate at a fixed 5% false positive rate. Our method, NA-PDD, significantly outperforms existing baseline methods in nearly all experimental settings. This performance advantage is particularly pronounced on the CCNewsPDD dataset. For instance, in the CCNewsPDD(mask) setting with the Pythia-2.8b model, NA-PDD achieves a TPR of 90.3%, which is more than double the performance of the strongest baseline, Probe Attack (44.3%). Likewise, on CCNewsPDD(prompt), its performance is generally more than twice that of the next-best methods. On the ArxivMIA and WikiMIA datasets, NA-PDD also demonstrates strong competitiveness, securing the top performance on most models. While individual baseline methods perform commendably in specific settings, our method exhibits superior and more consistent performance across all tested models and datasets. This confirms its robust detection capability in the critical low-FPR regime.

C Implementation Details

C.1 Data Split Configuration

For our experimental setup, we carefully partitioned our datasets to ensure reliable evaluation: (i) *Our Method.* We construct reference activation patterns on 200 samples, tune hyperparameter on a 200-sample validation set, and evaluate final performance on an 800-sample test set. (ii) *Probe Attack.* The probe attack uses 200 non-member samples as its training pool—half of these (100 samples) are employed to fine-tune the model. It then validates on 400 samples and reports results on the same 800-sample test set. (iii) *Other Baseline Methods.* All remaining baselines are evaluated directly on the 800-sample test set.

C.2 Technical Implementation

Neuronal Activation Determination. We register hook functions on the FFN activation functions across all Transformer layers to capture post-GELU/ReLU outputs, and declare a neuron activated whenever its output exceeds the activation threshold τ.

Neuronal Identity Discrimination. We construct reference activation patterns using 100 member and 100 non-member samples; experiments show this quantity sufficient for stable performance.

Hyperparameter Selection. After sensitivity analysis over $\tau \in [0, 2]$, $\alpha \in [1.2, 2.0]$, and $K \in$

Method	ArxivMIA		WikiMIA		CCNewsPDD(trans)		CCNewsPDD(mask)		CCNewsPDD(prompt)	
	TinyL.	OpenL.	Pythia	OPT	Pythia	OPT	Pythia	OPT	Pythia	OPT
Reference-free										
Loss Attack	5.1	5.6	13.7	11.4	6.3	6.5	22.5	15.0	7.8	7.3
Neighbor Attack	7.9	6.5	**18.2**	14.2	6.8	7.5	26.0	26.3	7.8	5.0
Min-K% Prob	4.5	5.1	16.9	15.0	8.5	9.5	28.5	28.5	6.3	6.5
Min-K++% Prob	5.1	6.1	13.8	12.2	10.3	17.3	18.3	20.8	6.5	10.3
DC-PDD	3.6	4.8	11.3	12.9	7.0	9.0	29.8	27.8	2.8	2.8
Reference-based										
Zlib Compression	2.5	3.5	17.3	14.4	14.3	14.3	28.8	19.5	4.8	5.5
Lowercased Text	4.3	6.3	10.1	9.1	7.8	7.5	10.0	9.3	9.0	8.3
Smaller Model	–	10.5	4.5	13.0	1.0	7.3	1.0	4.8	1.0	9.0
Probe Attack	7.5	7.4	16.7	15.4	19.3	44.5	44.3	21.8	29.5	16.0
Our Method										
NA-PDD	**10.1**	**10.5**	16.6	**23.9**	**60.8**	**57.5**	**90.3**	**72.3**	**63.5**	**60.0**

Table 3: TPR@5%FPR (%) across ArxivMIA, WikiMIA, and CCNewsPDD datasets. Higher is better. Best per column in **bold**.

$[1, 32]$, we recommend setting the activation threshold $\tau = 1.5$, the dominance coefficient $\alpha = 1.8$, and selecting $K = 5$ layers.

Computational Resources: Our experiment completed on a single NVIDIA A100 (80GB) GPU, with inference time approximately 30 minutes per model, varying by model size and dataset scale.

C.3 Text Transformation Examples

To ensure that the data maintains authentic linguistic properties while remaining unseen by the target LLMs during pretraining, we apply a controlled data-transformation pipeline. Table 4 then demonstrates how each transformation preserves semantic content while introducing targeted variations. In the examples, the **boldfaced** spans mark the transformed segments that differ from the original text.

C.4 Rigorous Quality Assessment

To assess the quality of our transformed texts, we perform two statistical comparisons (Table 5). The Member & Member comparison acts as a baseline, measuring the internal diversity between two random 300-sample halves of the original data. The Member & Nonmember comparison then pits one of these original halves against 300 transformed samples to measure transformation quality.

The Member & Nonmember scores for all methods closely track the Member & Member baseline. Both lexical similarity (n-gram Jaccard) and semantic similarity remain high and comparable to the baseline. Even the JS Divergence, a sensitive measure of vocabulary distribution, increases only slightly from its 0.3447 baseline. This consistency provides strong evidence that our transformations preserve the original text's statistical character without introducing noticeable machine-generated artifacts, confirming our benchmark's validity.

Method	Text Example
Original Text	*Laura Aldridge, RN, has been named director of the medical, surgical, and pediatric units at West Valley Medical Center. She holds a bachelor's degree in nursing from Northwest Nazarene University.*
BackTranslation	**Process:** English → French → English **Result:** *Laura Aldridge, RN, has been **appointed Director** of Medical, Surgical, and **Paediatric Units** at **the** West Valley Medical Center **and holds a Bachelor of Nursing degree** from **Nazarene University in the Northwest**.*
BERT_Mask	**Process:** Randomly mask 15% of tokens and generate alternatives **Result:** *Laura Aldridge, **MD**, has been named director of the medical and surgical and **support units** at West Valley Medical Center. She holds a bachelor's degree in nursing from Northwest Nazarene University.*
Prompt_Rewrite	**Process:** GPT-4o model with prompt "Rewrite: {text}" **Result:** *Laura Aldridge, RN, has been **appointed** director of the medical, surgical, and pediatric units at West Valley Medical Center. She **earned** a bachelor's degree in nursing from Northwest Nazarene University.*

Table 4: Data Transformation Examples

Method	Comparison Type	1-gram	2-gram	Similarity	JS Divergence
BackTranslation	Member & Member	0.0842	0.0031	0.0480	0.3447
	Member & Nonmember	0.0774	0.0032	0.0525	0.4222
BERT_Mask	Member & Member	0.0842	0.0031	0.0480	0.3447
	Member & Nonmember	0.0763	0.0027	0.0508	0.4193
Prompt_Rewrite	Member & Member	0.0842	0.0031	0.0480	0.3447
	Member & Nonmember	0.0724	0.0026	0.0570	0.4481

Table 5: Statistical analysis of text quality. The Member & Member comparison serves as a baseline comparing original texts against each other, while Member & Nonmember compares original texts to their transformed versions. For the similarity metrics (1-gram, 2-gram, Similarity), higher values indicate greater similarity. For JS Divergence, a lower value indicates that the vocabulary distributions are more similar.