SEAD: A Surrogate-free Label-only Membership Inference Attack against Pre-trained LLMs with Semantic-Aware Density

Biao Yi[1], **Jiahao Li**[2], **Yiming Li**[3*], **Yu He**[4], **Baolei Zhang**[1], **Zheli Liu**[1*], **Dacheng Tao**[3]
[1]College of Cyber Science, Nankai University [2]Xi'an Jiaotong University
[3]Nanyang Technological University [4]Zhejiang University
yibiao@mail.nankai.edu.cn

Abstract

Membership inference attacks (MIAs) aim to determine whether specific data was used to train a model. While existing MIAs against pre-trained Large Language Models (LLMs) typically require access to complete logits (probabilities), such access is sometimes unavailable in real-world deployments where only the generated text is exposed. Current label-only MIAs relied on surrogate models to estimate the target model's token probabilities, but we identify fundamental limitations: high sensitivity to surrogate model selection and significant probability estimation errors. To address these challenges, we propose SEAD (Semantic-Aware Density), a novel surrogate-free label-only MIA approach that directly estimates token probabilities through Monte Carlo sampling of the target model itself. This approach eliminates dependency on surrogate models while reducing probability estimation errors by an order of magnitude. Furthermore, we introduce a semantic-aware density approach that enhances attack effectiveness by considering both exact token matches and semantically similar alternatives, inspired by the understanding that LLMs may express memorized information through different but semantically equivalent tokens. Extensive evaluations demonstrate that SEAD consistently outperforms existing label-only attacks and serves as a foundational density estimator in the label-only setting[1].

1 Introduction

Large Language Models (LLMs) have achieved state-of-the-art results across diverse language tasks (Brown et al., 2020; Vaithilingam et al., 2022). However, their impressive capabilities also raise serious privacy concerns about the data used in pre-training (Ishihara, 2023; Shi et al., 2024). As a useful tool in auditing privacy risks, membership inference attacks (MIAs) (Shokri et al., 2017) can reveal whether specific data was included in a model's training set. It is also critical for auditing the unauthorized use of copyrighted data during pre-training (Shi et al., 2024; Zhang et al., 2025c) and verifying machine unlearning (Bourtoule et al., 2021; Yao et al., 2024).

Built on the philosophy that models tend to assign higher probabilities to member samples than to non-member samples, most prior MIAs against pre-trained LLMs assume a logits-based setting, where adversaries have access to the complete logits (probabilities) of generated token(s) (Carlini et al., 2021, 2022; Shi et al., 2024; Duan et al., 2024; Zhang et al., 2025c; Wang et al., 2025). However, this assumption is often violated in practice, as many real-world LLM APIs operate in a "label-only" setting, returning only generated text without the underlying probabilities. This renders conventional logit-based MIAs infeasible. Therefore, *we focus on the core problem of estimating token probabilities/density in the label-only setting*, thereby addressing the key challenge for MIA in practice.

While the exploration of this problem is still in its early stages, to the best of our knowledge, PETAL (He et al., 2025b) is the only existing work that specifically addresses token probability estimation in a label-only context. PETAL utilizes a publicly available language model as a surrogate to approximate the token probability of the target model and calculates an alternative perplexity to perform MIA. Specifically, it constructs a mapping between semantic similarity and token probability on the surrogate model via linear regression, and then applies this mapping to estimate the target model's probability assignments.

However, our research reveals that PETAL, as a representative of MIA methods requiring the surrogate model, has some fundamental limitations. Firstly, its performance **exhibits high sensitivity**

*Corresponding Author.
[1]Code is available at https://github.com/clearlove clearlove/SEAD

Findings of the Association for Computational Linguistics: ACL 2026, pages 6794–6813

to surrogate model selection. For example, we find that PETAL's average AUROC on Llama series models drops from 0.59 (with the default surrogate GPT2-XL) to 0.52 when using newer surrogates (e.g., Qwen2-7B). We attribute this degradation to *surrogate memorization bias*, where the surrogate's pre-training dataset potentially includes the target's non-members, leading to skewed PETAL's learned probability-similarity relationship. Consequently, effective surrogate selection becomes a challenging task: contrary to naive assumptions where a more powerful or contemporary surrogate might be preferred, it necessitates careful selection based on specific knowledge of model training histories, thus limiting PETAL's practical utility. Secondly, this approach **incurs substantial probability estimation errors**, defined as the discrepancies between the token probabilities estimated by the surrogate model and the true probabilities assigned by the target model. Such discrepancies are inherent when using a surrogate as an intermediary, given the inevitable gap between any surrogate and the target model. Our experiments show that even with carefully selected surrogate models, the average estimation error (measured by ℓ_1 distance) can reach more than 0.2, potentially leading to insufficient MIA effectiveness.

To overcome these limitations, we propose **SEAD** (**Se**mantic-**A**ware **D**ensity), a novel surrogate-free label-only MIA approach for pre-trained LLMs. Our key insight is that probability estimation can be performed by directly querying the target model via Monte Carlo sampling (Bishop, 2006), i.e., repeatedly sampling its next-token predictions, even without surrogate models. It reduces probability estimation errors by an order of magnitude with only 50 samples, with accuracy further improving as the sampling size increases. Moreover, given that probability estimation through sampling frequency alone may not fully capture the model's memorization behavior, we further introduce a semantic-aware density module to enhance attack effectiveness. This module aggregates the probability mass of both exact token matches and semantically similar alternatives, weighted by their semantic similarity to the target token. This approach exploits the property that LLMs may express memorized information through different but semantically equivalent tokens, providing a more robust membership inference signal than lexical matching alone. Based on these insights, we implement our SEAD in three key stages. Firstly, for each token position in the target sequence, we query the target model to collect a diverse set of next-token samples through Monte Carlo sampling; Secondly, we compute a semantic-aware density by aggregating the probability mass of all sampled tokens, weighting each by its semantic similarity to the target token as determined by a Natural Language Inference (NLI) model (He et al., 2021). This approach captures both exact matches and semantically equivalent alternatives that may indicate memorization; Thirdly, we compute a semantic-aware perplexity from these density values and infer membership via thresholding.

Our main contributions are four-fold. **(1)** We identify fundamental limitations in surrogate-based label-only MIAs for pre-trained LLMs, particularly their high sensitivity to surrogate model selection and significant probability estimation errors, both of which undermine attack reliability in practice. **(2)** We propose SEAD, a novel surrogate-free label-only MIA approach that directly estimates token probabilities through Monte Carlo sampling, thereby eliminating dependency on surrogate models and substantially reducing probability estimation errors. **(3)** We introduce a semantic-aware density approach that enhances attack effectiveness by aggregating the probability mass of both exact token matches and semantically similar alternatives, thus capturing memorization signals that go beyond lexical matching. **(4)** Extensive evaluations demonstrate that SEAD achieves state-of-the-art performance in the label-only setting and serves as a foundational density estimator that can be integrated with advanced calibration strategies to construct more powerful label-only attacks.

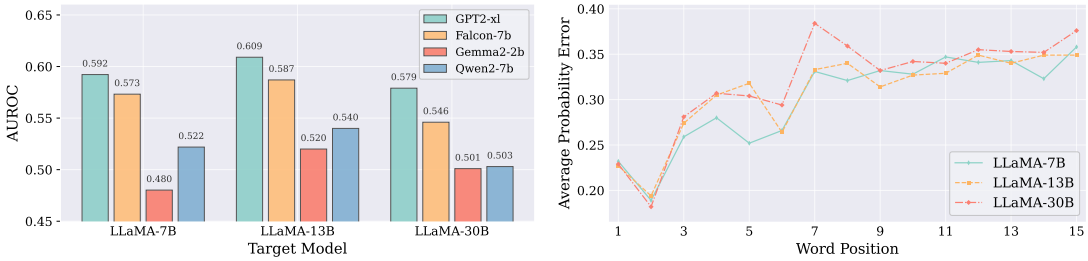
2 Revisiting Label-only MIAs

2.1 Preliminaries

In the context of LLMs, we follow the initial definition of MIAs by Shokri et al. (2017). Given an LLM f_θ parameterized by weights θ and its pre-training dataset D sampled from the underlying distribution π, f_θ is solely pre-trained on D. For any target point $x_i \in \pi$, the adversary employs an attack method A to predict its membership state m_i, where $m_i = 1$ if $x_i \in D$; otherwise $m_i = 0$. We can thus formally define A as follows:

$$A(x_i, \theta) = \mathbb{1}[S(m_i = 1 | x_i, \theta) \geq \tau], \quad (1)$$

where S represents a score function that quantifies the likelihood of x_i being a member, and τ indi-



(a) Sensitivity to Surrogate Model Selection

(b) Probability Estimation Errors

Figure 1: Limitations of the surrogate-based MIA method PETAL. (a) Attack performance (AUROC) of PETAL is highly sensitive to the choice of surrogate model. (b) PETAL incurs substantial probability estimation errors.

cates a threshold used for decision-making. The adversary does not have any knowledge of D.

The foundation of many MIAs against LLMs is that due to their maximum likelihood estimation training, models tend to assign higher probabilities to members than to non-members (Carlini et al., 2022; Watson et al., 2021). This phenomenon manifests in many attacks such as PPL attack (Carlini et al., 2021), Min-K% (Shi et al., 2024), Min-K%++ (Zhang et al., 2025c). Taking the PPL attack as an example, the score function for a given sequence $x = (t_1, \ldots, t_n)$ is defined as:

$$\begin{aligned} S(x) &= -\text{PPL}(x) \\ &= -\exp\left(-\frac{1}{n}\sum_{i=1}^{n}\log p(t_i|t_{<i})\right) \end{aligned} \tag{2}$$

Adversary's Capabilities. In a label-only setting, the adversary's information is limited to only the sequence of generated tokens for any input, without access to the probabilities assigned to each token. A detailed discussion is provided in Appendix B.

Surrogate-based Label-only MIA for Pre-trained LLMs. Conventional MIAs rely heavily on token probabilities, which are inaccessible in a label-only setting. PETAL (He et al., 2025b) leverages an alternative, publicly available language model to estimate the token probability of the target model. The key insight behind PETAL is the observation that the probability of a particular token being generated by the target model is potentially correlated with the semantic similarity between that token and the token actually generated by the model. PETAL utilizes a surrogate model (e.g., an open-source LLM) to collect semantic similarity-probability pairs for all tokens in target sequences, then performs a univariate linear regression to establish a mapping between semantic similarity and output probabilities:

$$\log p(t_i|t_{<i}) = \beta \cdot \text{sim}(t_i, \hat{t}_i) + \alpha, \tag{3}$$

where $\hat{t}_i$ is the token generated by the model when prompted with the prefix $t_{<i}$. For the target model, PETAL queries to get token-level semantic similarity scores, approximates the probabilities assigned by the target model using the regression model, and calculates an alternative perplexity.

2.2 Limitations of Surrogate-based MIA

Despite its innovative design, we reveal that PETAL exhibits two significant limitations that affect its reliability and effectiveness.

Limitation 1: Sensitivity to Surrogate Model Selection. PETAL's performance critically depends on the selection of an appropriate surrogate model. We conducted experiments using different surrogate models (GPT2-XL (Radford et al., 2019) (the default used in PETAL), Falcon-7B (Almazrouei et al., 2023), Gemma-2-2B (Team et al., 2024), and Qwen2-7B (Yang et al., 2024)) to attack Llama series models (Touvron et al., 2023) on the WikiMIA dataset (Shi et al., 2024). Figure 1a shows the Area Under the Receiver Operating Characteristic Curve (AUROC) scores across these different surrogate configurations. The results show a dramatic performance drop when switching from GPT2-XL (average AUROC: 0.59) and Falcon-7B (average AUROC: 0.57) to Gemma-2-2B (average AUROC: 0.50) and Qwen2-7B (average AUROC: 0.52).

This sensitivity appears linked to what we term "surrogate memorization bias", which can arise when the surrogate model's training data significantly overlaps with the MIA benchmark. The WikiMIA dataset was designed specifically for models released between 2017 and 2023. Models like GPT-2 XL and Falcon-7B align with this period, and the probability-similarity relationship learned by PETAL from these surrogates might therefore generalize more effectively to Llama on this specific dataset. In contrast, newer models (e.g., Gemma-2-2B, Qwen2-7B, released 2024)

may have been pre-trained on substantial portions of the WikiMIA dataset, critically including samples that are "non-members" for the target model (Llama). Extensive training of a surrogate on such benchmark data leads to its "memorization" of these samples. This, in turn, skews the probability-similarity relationship that PETAL derives from the surrogate. When this potentially biased relationship is then used to estimate probabilities for the target model, it may not accurately capture the target's true probability distribution, particularly in distinguishing its actual members from non-members.

Limitation 2: Large Probability Estimation Errors. The linear mapping between semantic similarity and token probability introduces substantial estimation errors. To quantify these errors, we measured the ℓ_1 distance between the true probabilities and the PETAL-estimated probabilities across various token positions in sequences from WikiMIA dataset. Figure 1b illustrates the average ℓ_1 distance when estimating probabilities for the Llama series models, where the surrogate model is GPT2-XL. The results reveal consistently large estimation errors, with average ℓ_1 distances ranging from approximately 0.2 to 0.4 across different token positions. Moreover, there is a trend of increasing error as the token position index increases. These substantial errors suggest that PETAL's probability approximation mechanism may not accurately capture the target model's true probability distribution, potentially undermining the attack's effectiveness.

These findings highlight fundamental limitations of surrogate-based approaches such as PETAL. First, their effectiveness is critically dependent on surrogate model selection. The surrogate memorization bias reveals a counter-intuitive reality: merely selecting a more powerful or contemporary surrogate does not ensure optimal attack performance. Instead, avoiding skewed estimations requires a careful, specific assignment of the surrogate with the target model and benchmark, a nuanced task that is difficult for an adversary to perform in practice without detailed knowledge of both models' training histories. Moreover, results show that even when using a carefully-selected surrogate model, the probability estimation errors remain substantial. The cause of these high estimation errors lies in the inherent limitations of using a surrogate as an intermediary, given the inevitable gap between any surrogate and the target model. These limitations motivate the development of a surrogate-free, label-only attack that can overcome these issues and provide more robust MIA.

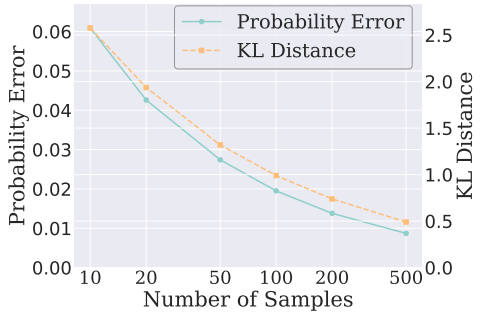


Figure 2: Probability error and KL distance decrease with increasing sampling.

3 Methodology

In this section, we present SEAD (Semantic-Aware Density), a novel surrogate-free label-only MIA for pre-trained LLMs. SEAD fundamentally removes the need for surrogate models by directly estimating token probabilities through Monte Carlo sampling (Bishop, 2006). To further improve performance, SEAD introduces a semantic-aware density measure that aggregates probability mass over semantically similar tokens, thus better capturing the true memorization signal of the model.

3.1 Monte Carlo Probability Estimation

Instead of relying on a surrogate model to approximate token probabilities, we propose a direct estimation approach using Monte Carlo sampling. Given a sequence $x = t_1, t_2, ..., t_n$, our goal is to estimate the probability $p(t_i|t_{<i})$ for each token t_i in the sequence. For each prefix $t_{<i} = t_1, \ldots, t_{i-1}$, we obtain K predictions for the next token by querying the target model. Modern LLM APIs allow generating K samples via a single query, which is critical for our method as it avoids the prohibitive overhead associated with making K separate API requests. Let $t_i^{(1)}, t_i^{(2)}, ..., t_i^{(K)}$ be these sampled tokens. We estimate the probability of the original token t_i as:

$$\hat{p}(t_i|t_{<i}) = \max\left(\epsilon, \frac{1}{K}\sum_{j=1}^{K}\mathbb{1}[t_i^{(j)} = t_i]\right), \quad (4)$$

where $\mathbb{1}[\cdot]$ is the indicator function that equals 1 when the condition is true and 0 otherwise, and ϵ is a small positive constant (e.g., 10^{-12}) to ensure numerical stability for logarithmic operations.

Unlike surrogate-based PETAL which requires a carefully selected surrogate model, our method

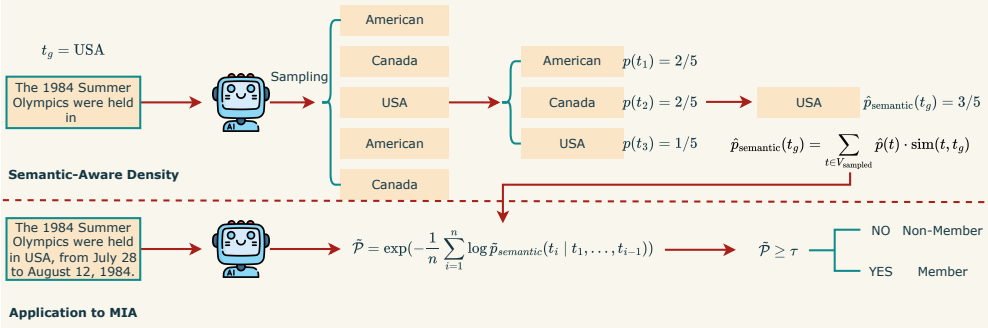


Figure 3: Overview of the SEAD attack pipeline. (1) Next-token probabilities are directly estimated from the target LLM using Monte Carlo sampling. (2) A novel semantic-aware density is then computed by considering both exact matches and semantically similar sampled tokens (e.g., “America" for target “USA"), weighted by similarity. This density informs a semantic-aware perplexity score for membership inference.

estimates token probabilities directly from the target model’s outputs, fundamentally resolving the surrogate selection sensitivity issue that plagues prior methods. Moreover, our empirical results in Figure 2 show that with just 50 samples, the average probability estimation error drops to approximately 0.03, which is an order of magnitude lower than the typical errors (0.2–0.4) observed in surrogate-based PETAL. As the number of samples increases, the estimation accuracy continues to improve, for example, when the number of samples reaches 500, the error drops to below 0.01 (another order of magnitude lower), providing a significantly more precise and robust foundation for membership inference.

3.2 Semantic-Aware Density

Directly estimating token probabilities via Monte Carlo sampling provides a more accurate empirical approximation of the model’s output distribution. However, relying solely on the frequency of exact token matches (which we term **lexical density**) is often insufficient to capture the model’s true memorization behavior. Natural language is inherently redundant and highly expressive; multiple lexical forms can express the same underlying semantic meaning. For instance, given the prompt “The 1984 Summer Olympics were held in”, both “USA” and “America” are lexically distinct but semantically similar, and either completion indicates the model’s memorization of this factual content. A model that has memorized “USA” may just as likely generate “America” if prompted.

To obtain a more faithful and robust MIA signal, we propose **semantic-aware density**. Instead of only considering exact lexical matches, this metric aggregates evidence from all tokens sampled from the model, weighting each by its semantic similarity to the actual target token t_i. This approach considers the total probability mass assigned to tokens semantically close to the target.

Formal Definition of Semantic-Aware Density. Given a target sequence $x = (t_1, \ldots, t_n)$, for each position i and corresponding target token t_i, we define the *semantic-aware density* as:

$$p_{\text{semantic}}(t_i \mid t_{<i}) = \sum_{t \in V_{\text{sampled}}} \hat{p}(t \mid t_{<i}) \cdot \text{sim}(t, t_i) \tag{5}$$

where V_{sampled} is the set of unique tokens obtained from K Monte Carlo samples for the next token given the context $t_{<i}$. $p(t \mid t_{<i})$ is the empirical probability of token t under Monte Carlo sampling. $\text{sim}(t, t_i)$ is a score (typically between 0 and 1) quantifying the semantic similarity between a sampled token t and the target token t_i. In general, this formulation captures partial memorization evidence even when the model outputs tokens semantically close to t_i but not exact matches. In contrast to lexical density (which considers only $p(t_i \mid \ldots)$), semantic-aware density yields a more robust signal by aggregating contributions from all semantically related tokens in the samples. Given $\sum_{t \in V_{\text{sampled}}} p(t \mid \ldots) = 1$ and $\text{sim}(t, t_i) \in [0, 1]$, the resultant p_{semantic} also a bounded score (≤ 1), offering a richer indicator of the model’s inclination towards the meaning of t_i.

NLI-based Semantic Similarity. To quantify semantic similarity between tokens, we utilize Natural Language Inference (NLI) models (He et al., 2021), which are trained to determine the relationship between text pairs. For each sampled token $\tilde{t}_j$ and the target token t_i, we compute an entailment score to assess their semantic relatedness:

$$sim(t_i, \tilde{t}_j) = \begin{cases} 1 & \text{if } t_i \text{ entails } \tilde{t}_j \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

This binary scoring function treats sampled tokens that are semantically entailed by the target token as positive evidence for membership, while treating contradicting or neutral tokens as negative evidence. Unlike general embedding similarity, NLI models provide directional entailment relationships, which are particularly valuable for MIA scenarios. For example, given the prompt "The 1984 Summer Olympics were held in", using embedding similarity, the sampled token "Canada" and the ground truth token "USA" receive a high similarity score of 0.77 which is close to "America" and "USA" at 0.84, simply because both are country names and often appear in similar contexts. However, from a memorization standpoint, "America" is factually correct and reflects genuine model memorization, while "Canada" does not. Embedding-based similarity thus overestimates semantic equivalence between such tokens. In contrast, leveraging NLI models can accurately distinguish this difference by identifying that the ground truth token "USA" entails the factually correct token "America", whereas it does not entail "Canada", thereby providing a more reliable signal of memorization for MIA.

Membership Inference based on Semantic-Aware Perplexity. To infer membership, the adversary can compute the *semantic-aware perplexity* based on the $p_{semantic}(t)$ obtained from Equation (5) as:

$$\tilde{\mathcal{P}} = \exp(-\frac{1}{n}\sum_{i=1}^{n}\log p_{semantic}(t_i \mid t_{<i})). \quad (7)$$

Since the model will assign a lower semantic-aware perplexity to the member sequence, we can detect them by thresholding this result.

4 Experiments

4.1 Experimental Settings

Benchmarks. We evaluate our method on two standard benchmarks for MIAs against pre-trained LLMs: WikiMIA (Shi et al., 2024) and MIMIR (Duan et al., 2024). WikiMIA, the first benchmark designed for pre-training data detection, sources texts from Wikipedia events and labels them as member or non-member based on the event date relative to a model's release date. MIMIR is constructed from the training and test splits of the Pile dataset (Gao et al., 2021) to reduce the impact of distributional factors. We acknowledge the observation by Duan et al. (2024) that WikiMIA exhibits distributional and temporal discrepancies (Zhang et al., 2025b; Das et al., 2025; Maini et al., 2024; Meeus et al., 2024b) between its member and non-member sets. Nevertheless, we argue that identifying membership against temporally shifted non-members presents a practical and relevant challenge for LLMs trained on evolving web data. Therefore, following prior work (Zhang et al., 2025c; Xie et al., 2024; He et al., 2025b), we report results on both benchmarks to ensure experimental integrity.

Models. For WikiMIA, we evaluate on a diverse set of publicly available LLMs: Mamba (1.4B) (Gu and Dao, 2023), OPT (6.7B) (Zhang et al., 2022), Pythia (6.9B) (Biderman et al., 2023), GPT-NeoX-20B (Black et al., 2022), and LLaMA (13B, 30B) (Touvron et al., 2023). WikiMIA is suitable for these models as they were trained on Wikipedia data and their time cutoffs align with WikiMIA's settings. For MIMIR, we focus on Pythia models (1.4B, 2.8B, 6.9B, 12B), consistent with Duan et al. (2024), as these models are pre-trained on the Pile.

Baselines. We compare SEAD with state-of-the-art MIAs: ***Logits-based attacks:*** We include the PPL attack (Yeom et al., 2018; Carlini et al., 2021), Reference attack (Carlini et al., 2021), Zlib attack (Carlini et al., 2021), Neighborhood attack (Mattern et al., 2023), Mink (Shi et al., 2024), Mink++ (Zhang et al., 2025c), and RECALL (Xie et al., 2024). ***Label-only attacks:*** PETAL(He et al., 2025b) serves as our primary comparison point.

Implementation Details. When evaluating our SEAD, we use `deberta-xlarge-mnli` (He et al., 2021) as the NLI model to compute the semantic-aware density. The number of Monte Carlo samples is set to 50 by default, and following the default settings of many current closed-source LLMs, the temperature coefficient is set to 0.7. All the experiments are done with an A800-80G.

Metrics. The primary evaluation metric is the Area Under the Receiver Operating Characteristic Curve (*AUROC*) (Carlini et al., 2021; Shi et al., 2024), which is threshold-independent and reflects the probability that a randomly chosen member sample is assigned a higher score than a non-member. In addition to AUROC, we also report the True Positive Rate (*TPR*) at low False Positive Rate (*FPR*) thresholds to evaluate the practical effectiveness of the attacks in low-false-positive regimes.

Table 1: AUROC results on WikiMIA benchmark with bold indicating best label-only attack.

		Logits-based Attacks							*Label-only Attacks*	
Len	**Models**	**PPL**	**Ref**	**Zlib**	**Neighbor**	**Mink**	**Mink++**	**RECALL**	**PETAL**	**SEAD**
32	**Mamba-1.4B**	0.60	0.60	0.61	0.58	0.63	0.68	0.91	0.60	**0.65**
	Opt-6.7B	0.59	0.59	0.60	0.58	0.61	0.66	0.80	0.60	**0.62**
	Pythia-6.9B	0.62	0.58	0.63	0.61	0.66	0.71	0.91	0.62	**0.67**
	Llama-13B	0.66	0.59	0.67	0.62	0.66	0.85	0.90	0.62	**0.81**
	Neox-20B	0.68	0.53	0.68	0.65	0.71	0.77	0.90	0.66	**0.74**
	Llama-30B	0.69	0.59	0.69	0.64	0.69	0.85	0.93	0.63	**0.79**
64	**Mamba-1.4B**	0.58	0.65	0.60	0.61	0.62	0.67	0.85	0.60	**0.64**
	Opt-6.7B	0.57	0.64	0.60	0.61	0.60	0.65	0.81	0.60	**0.61**
	Pythia-6.9B	0.61	0.61	0.63	0.63	0.65	0.72	0.90	0.61	**0.69**
	Llama-13B	0.64	0.62	0.65	0.64	0.66	0.84	0.94	0.62	**0.78**
	Neox-20B	0.67	0.56	0.68	0.67	0.72	0.77	0.90	0.66	**0.74**
	Llama-30B	0.66	0.61	0.67	0.67	0.68	0.84	0.94	0.63	**0.79**
128	**Mamba-1.4B**	0.63	0.69	0.66	0.65	0.67	0.68	0.89	**0.66**	0.64
	Opt-6.7B	0.63	0.68	0.64	0.64	0.67	0.69	0.74	**0.66**	**0.66**
	Pythia-6.9B	0.65	0.64	0.68	0.68	0.70	0.70	0.85	0.67	**0.69**
	Llama-13B	0.68	0.66	0.70	0.68	0.72	0.84	0.91	0.66	**0.77**
	Neox-20B	0.71	0.59	0.72	0.72	0.76	0.75	0.86	0.69	**0.72**
	Llama-30B	0.70	0.65	0.72	0.72	0.74	0.83	0.92	0.68	**0.76**
	Average	0.64	0.62	0.66	0.64	0.68	0.75	0.88	0.64	**0.71**

Table 2: AUROC results on MIMIR benchmark with bold indicating best label-only attack.

	Wikipedia				**GitHub**				**Pile CC**				**PubMed Central**			
Method	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B
Logits-based Attacks:																
PPL	0.54	0.54	0.54	0.55	0.65	0.66	0.66	0.67	0.49	0.49	0.49	0.50	0.48	0.47	0.49	0.48
Ref	0.50	0.50	0.50	0.50	0.58	0.59	0.59	0.59	0.50	0.50	0.50	0.50	0.50	0.50	0.49	0.49
Zlib	0.54	0.54	0.54	0.56	0.66	0.67	0.67	0.68	0.48	0.48	0.49	0.49	0.48	0.47	0.48	0.48
Neighbor	0.56	0.56	0.57	0.58	0.66	0.67	0.68	0.69	0.51	0.52	0.52	0.53	0.53	0.53	0.52	0.55
Mink	0.54	0.55	0.56	0.58	0.64	0.66	0.66	0.67	0.50	0.50	0.51	0.51	0.48	0.47	0.48	0.49
Mink++	0.55	0.55	0.55	0.58	0.56	0.60	0.59	0.61	0.53	0.53	0.55	0.55	0.45	0.45	0.46	0.46
RECALL	0.53	0.54	0.51	0.55	0.66	0.68	0.68	0.69	0.51	0.53	0.53	0.53	0.49	0.48	0.50	0.50
Label-only Attacks:																
PETAL	0.54	0.52	0.52	0.53	**0.65**	0.65	0.65	**0.66**	0.49	0.49	**0.49**	**0.49**	**0.48**	0.47	**0.48**	0.47
SEAD	**0.55**	**0.55**	**0.56**	**0.57**	0.64	**0.66**	**0.67**	**0.66**	**0.50**	**0.52**	0.47	**0.49**	**0.48**	**0.48**	**0.48**	**0.48**

	ArXiv				**DM Mathematics**				**HackerNews**				**Average**			
Method	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B
Logits-based Attacks:																
PPL	0.51	0.52	0.53	0.53	0.48	0.48	0.49	0.48	0.52	0.53	0.53	0.53	0.52	0.53	0.53	0.54
Ref	0.45	0.45	0.45	0.45	0.49	0.48	0.47	0.47	0.54	0.54	0.54	0.53	0.49	0.49	0.49	0.49
Zlib	0.51	0.52	0.52	0.52	0.48	0.47	0.48	0.48	0.52	0.52	0.52	0.53	0.52	0.52	0.53	0.53
Neighbor	0.56	0.57	0.58	0.58	0.51	0.52	0.53	0.53	0.54	0.56	0.57	0.57	0.55	0.56	0.57	0.57
Mink	0.49	0.51	0.51	0.51	0.48	0.48	0.50	0.50	0.52	0.52	0.53	0.53	0.52	0.53	0.54	0.54
Mink++	0.49	0.51	0.52	0.52	0.51	0.50	0.51	0.52	0.54	0.51	0.54	0.54	0.52	0.52	0.53	0.54
RECALL	0.52	0.53	0.50	0.52	0.52	0.53	0.55	0.54	0.51	0.53	0.52	0.53	0.54	0.54	0.54	0.55
Label-only Attacks:																
PETAL	0.48	0.50	0.49	0.49	0.47	0.48	0.48	0.49	**0.52**	**0.52**	**0.52**	**0.53**	**0.52**	0.52	0.52	0.52
SEAD	**0.49**	**0.51**	**0.51**	**0.53**	**0.49**	**0.49**	**0.50**	**0.50**	0.50	**0.52**	**0.52**	**0.53**	**0.52**	**0.53**	**0.53**	**0.54**

4.2 Main Results

State-of-the-art Performance in Label-only Setting. As shown in Table 1-2, SEAD consistently establishes new state-of-the-art performance within the label-only setting. On the WikiMIA benchmark, SEAD achieves an average AUROC of 0.71, significantly outperforming the baseline PETAL (0.64) and achieving the highest label-only AUROC in 17 of the 18 evaluated attack configurations. This trend of improved performance extends to the MIMIR benchmark, which is known to be particularly challenging: even logits-based methods often yield modest results ($i.e.$, with AUROC only slightly higher than 0.5) on this dataset. On the MIMIR benchmark, SEAD achieves an average AUROC of 0.53, outperforming PETAL (0.52), and attains the best label-only AUROC in 25 of the 28 evaluated attack configurations. To further evaluate its practical efficacy, we also report the TPR@5%FPR in Section C.1. Since both methods ultimately compute a perplexity-like score based on estimated token densities, **this consistent outperformance demonstrates SEAD's core advantage as a superior label-only density estimator compared to PETAL.**

Comparable Performance to Several Logits-based Attacks. Despite the information constraints of the label-only setting, SEAD's performance is comparable to several established logits-based MIAs. On WikiMIA, its average AUROC of 0.71 outperforms several logits-based methods such as PPL (0.64) and Mink (0.68). On the MIMIR benchmark, SEAD's average AUROC of 0.53 remains

competitive and on par with several logits-based methods, including PPL (0.53) and Mink++ (0.53).

PPL directly utilizes the ground-truth token probabilities, which serves as a powerful logits-based baseline for density estimation. The fact that SEAD can consistently match or even surpass this baseline's performance across different benchmarks is a direct testament to its effectiveness as a label-only density estimator. Furthermore, SEAD's utility extends beyond a standalone attack. As detailed in Appendix C.2, **its density estimation can be integrated with advanced calibration strategies like RECALL to achieve even stronger label-only performance**, for example, AUROC improves from 0.63 to 0.75 on Mamba-1.4B and from 0.66 to 0.80 on Pythia-6.9B. This underscores SEAD's potential as a foundational density estimator component for advanced label-only MIAs.

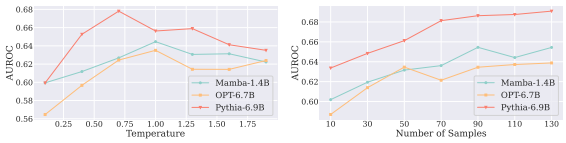
4.3 Ablation Study

Semantic-Aware vs. Lexical Density. Here, we further investigate how the semantic-aware density compares to its Monte Carlo lexical density counterpart. As shown in Table 3, the AUROC performance of both density methods on different victim models is reported. Semantic-aware density consistently outperforms lexical density across all victim models. Notably, for `OPT-6.7B` and `Pythia-6.9B`, the improvement is approximately 3%, while for `Mamba-1.4B`, the enhancement reaches 6%. These results demonstrate that the semantic-aware density method proposed in this paper effectively improves membership inference attack performance compared to lexical density. Moreover, as detailed in Appendix C.4, this enhancement is versatile and also significantly boosts the lexical density calculated with ground-truth logits.

The Influence of Different Temperature Coefficients. Experimental results illustrated in Figure 4a show that the performance of MIA deteriorates when the temperature coefficient τ is either too large or too small, achieving the best performance when the temperature is close to 1. This is because excessively high or low temperature coefficients distort the original output distribution, causing the probability estimated via sampling to deviate from the true probability, and thus resulting in reduced performance. Within the moderate range $\tau \in [0.7, 1.3]$, AUROC varies only slightly, indicating robustness to typical provider defaults. Most production APIs (e.g., GPT-series) expose a temperature knob, so SEAD remains practical in real deployments.

Table 3: AUROC comparison between lexical density and semantic-aware density methods.

Methods	Mamba-1.4B	OPT-6.7B	Pythia-6.9B
Lexical Density	0.59	0.60	0.63
Semantic-Aware Density	0.65	0.63	0.66



(a) Temperature Coefficients (b) Sample Numbers

Figure 4: AUROC of SEAD with different temperature coefficients and sample numbers.

The Influence of Different Sample Numbers. We here evaluated the AUROC of SEAD as the number of samples varied from 10 to 130. As shown in Figure 4b, the performance of SEAD gradually improves as the number of samples increases. This aligns with intuition, as simulating token probabilities becomes more accurate with larger numbers of samples, thereby enhancing the MIA performance.

Further evaluations, with detailed results available in the appendix. Our ablation studies (Appendix C.3) justify our choice of NLI over cosine similarity and demonstrate SEAD's robust performance across various NLI models. Moreover, experiments on the paraphrased WikiMIA benchmark (Appendix C.5) indicate that SEAD's attack effectiveness is maintained even when input text is paraphrased. Our efficiency analysis (Appendix C.6) shows SEAD provides a better accuracy-efficiency trade-off over PETAL. A case study on a representative closed-source LLM was also conducted (Appendix C.7), further highlighting SEAD's superior performance compared to PETAL.

5 Conclusion

In this work, we proposed SEAD, a novel surrogate-free label-only MIA for pre-trained LLMs. This method is motivated by our analyses showing that existing label-only MIA methods have significant limitations, including high sensitivity to surrogate model selection and large probability estimation errors. We implemented SEAD through a direct Monte Carlo sampling approach that eliminates the need for surrogate models while reducing estimation errors by an order of magnitude. We also designed a semantic-aware density that enhances attack effectiveness by considering both exact token matches and semantically similar alternatives, ex-

ploiting the property that LLMs may express memorized information through different but semantically equivalent tokens. Results on benchmark datasets demonstrated that SEAD outperforms existing label-only attacks and serves as a foundational density estimator in the label-only setting.

Limitations

In this section, we discuss the potential limitations and future directions of our work.

Firstly, the label-only setting, by its nature, presents inherent information constraints compared to logits-based scenarios. While SEAD significantly advances capabilities within this setting, the fundamental trade-off between efficiency budget and estimation accuracy via Monte Carlo sampling remains a characteristic of such approaches. However, as shown in our efficiency analysis (Section C.6), we demonstrated that SEAD can be configured with fewer Monte Carlo samples to achieve inference speeds faster than existing label-only methods while maintaining comparable attack performance. Future work could explore novel theoretical frameworks to further optimize this trade-off or investigate alternative information extraction techniques beyond direct probability estimation, while still adhering to label-only constraints.

Secondly, our current investigation, like many in the field, primarily focuses on textual LLMs. The landscape of generative AI is rapidly expanding to include multimodal models. Extending membership inference paradigms, including semantic-aware density concepts, to these more complex architectures presents a significant and exciting research challenge. This would necessitate developing cross-modal semantic understanding and adapting attack strategies to account for how information from different modalities might be memorized and expressed. Further research could also delve into more abstract forms of memorization that go beyond direct semantic similarity of generated tokens.

Ethical Considerations

Our work contributes to the tools available for auditing LLM privacy, particularly in label-only access scenarios. By demonstrating a more effective surrogate-free attack, we aim to raise awareness within the AI community about the ongoing risks of membership inference, even when model access is restricted. It is crucial to emphasize that this research is intended to highlight vulnerabilities for defensive purposes and to encourage the development of more robust privacy-preserving measures; we do not endorse the malicious application of such attacks. Understanding these risks is a necessary step towards building more secure and trustworthy AI systems, and we hope our work motivates further investigation into LLM memorization and the creation of effective safeguards.

The challenge of data memorization in LLMs and the associated MIA risks are significant and not easily resolved. Therefore, we strongly advocate for LLM developers to exercise caution when including potentially sensitive information in pre-training datasets and to actively implement and evaluate defensive techniques. The datasets used in our study, WikiMIA and MIMIR, are publicly available benchmarks established for MIA research, and our use is confined to these research objectives. We are releasing the code for SEAD to facilitate further research, including the development of stronger defenses and a better understanding of the vulnerabilities inherent in LLMs.

Acknowledgments

This work was supported in part by the Joint Fund of the National Natural Science Foundation of China (No. U25B2028), and in part by the Key Program of the National Natural Science Foundation of China (No. 62432012). Dr Tao's research is partially supported by NTU RSR and Start Up Grants.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *arXiv:2311.16867*.

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 *ICML*.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer.

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. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv:2204.06745*.

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)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *NeurIPS*.

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

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)*.

Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *ICML*.

Debeshee Das, Jie Zhang, and Florian Tramèr. 2025. Blind baselines beat membership inference attacks for foundation models. *arXiv preprint arXiv:2406.16201*.

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

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2025. Mia-tuner: Adapting large language models as pre-training text detector. In *AAAI*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Jianing Geng, Biao Yi, Zekun Fei, Tongxi Wu, Lihai Nie, and Zheli Liu. 2025. When safety detectors aren't enough: A stealthy and effective jailbreak attack on llms via steganographic techniques. *arXiv preprint arXiv:2505.16765*.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*.

Jinwen He, Yiyang Lu, Zijin Lin, Kai Chen, and Yue Zhao. 2025a. Privacyxray: Detecting privacy breaches in llms through semantic consistency and probability certainty. *arXiv preprint arXiv:2506.19563*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *ICLR*.

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

Ruihan Hu, Yu-Ming Shang, Jiankun Peng, Wei Luo, Yazhe Wang, and Xi Zhang. 2025. Automated detection of pre-training text in black-box llms. In *IJCAI*.

Zhiheng Huang, Yannan Liu, Daojing He, and Yu Li. 2025. Df-mia: A distribution-free membership inference attack on fine-tuned large language models. In *AAAI*.

Shotaro Ishihara. 2023. Training data extraction from pre-trained language models: A survey. In *The Third Workshop on Trustworthy Natural Language Processing*.

Gyuwan Kim, Yang Li, Evangelia Spiliopoulou, Jie Ma, Miguel Ballesteros, and William Yang Wang. 2024. Detecting training data of large language models via expectation maximization. *arXiv preprint arXiv:2410.07582*.

Klas Leino and Matt Fredrikson. 2020. Stolen memories: Leveraging model memorization for calibrated {White-Box} membership inference. In *29th USENIX security symposium (USENIX Security 20)*.

Marvin Li, Jason Wang, Jeffrey G. Wang, and Seth Neel. 2023. Mope: Model perturbation based privacy attacks on language models. In *EMNLP*.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2024. Backdoor learning: A survey. *IEEE Trans. Neural Networks Learn. Syst.*

Zheng Li and Yang Zhang. 2021. Membership leakage in label-only exposures. In *CCS*.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024. LLM dataset inference: Did you train on my dataset? In *NeurIPS*.

Disha Makhija, Manoj Ghuhan Arivazhagan, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. 2025. Neural breadcrumbs: Membership inference attacks on llms through hidden state and attention pattern analysis. *arXiv preprint arXiv:2509.05449*.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. In *Findings of ACL*.

Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. 2024a. Did the neurons read your book? document-level membership inference for large language models. In *USENIX Security*.

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

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022. Quantifying privacy risks of masked language models using membership inference attacks. In *EMNLP*.

Hamid Mozaffari and Virendra J. Marathe. 2024. Semantic membership inference attack against large language models. *arXiv preprint arXiv:2406.10218*.

Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*.

Ashwinee Panda, Xinyu Tang, Christopher A. Choquette-Choo, Milad Nasr, and Prateek Mittal. 2025. Privacy auditing of large language models. In *ICLR*.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *ICLR*.

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

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. In *ICML*.

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

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *ICLR*.

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)*.

Hongyi Tang, Zhihao Zhu, and Yi Yang. 2025. Identifying pre-training data in llms: A neuron activation-based detection framework. *arXiv preprint arXiv:2507.16414*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv:2302.13971*.

Priyan Vaithilingam, Tianyi Zhang, and Elena L Glassman. 2022. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *Chi conference on human factors in computing systems extended abstracts*.

Cheng Wang, Yiwei Wang, Bryan Hooi, Yujun Cai, Nanyun Peng, and Kai-Wei Chang. 2025. Con-recall: Detecting pre-training data in llms via contrastive decoding. In *COLING*.

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

Yutong Wu, Han Qiu, Shangwei Guo, Jiwei Li, and Tianwei Zhang. 2024. You only query once: An efficient label-only membership inference attack. In *ICLR*.

Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. 2024. ReCaLL: Membership inference via relative conditional log-likelihoods. In *EMNLP*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report. *arXiv:2407.10671*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024. Large language model unlearning. In *NeurIPS*.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2022. Enhanced membership inference attacks against machine learning models. In *CCS*.

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)*.

Biao Yi, Sishuo Chen, Yiming Li, Tong Li, Baolei Zhang, and Zheli Liu. 2024. Badacts: A universal backdoor defense in the activation space. In *Findings of ACL*.

Biao Yi, Zekun Fei, Jianing Geng, Tong Li, Lihai Nie, Zheli Liu, and Yiming Li. 2025a. Badreasoner: Planting tunable overthinking backdoors into large reasoning models for fun or profit. *arXiv preprint arXiv:2507.18305*.

Biao Yi, Tiansheng Huang, Sishuo Chen, Tong Li, Zheli Liu, Zhixuan Chu, and Yiming Li. 2025b. Probe before you talk: Towards black-box defense against backdoor unalignment for large language models. In *ICLR*.

Biao Yi, Tiansheng Huang, Baolei Zhang, Tong Li, Lihai Nie, Zheli Liu, and Li Shen. 2025c. CTRAP: embedding collapse trap to safeguard large language models from harmful fine-tuning. *arXiv preprint arXiv:2505.16559*.

Biao Yi, Jiahao Li, Baolei Zhang, Lihai Nie, Tong Li, Tiansheng Huang, and Zheli Liu. 2025d. Gradient surgery for safe LLM fine-tuning. *arXiv preprint arXiv:2508.07172*.

Hengxiang Zhang, Songxin Zhang, Bingyi Jing, and Hongxin Wei. 2025a. Fine-tuning can help detect pretraining data from large language models. In *ICLR*.

Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. 2025b. Position: Membership inference attacks cannot prove that a model was trained on your data. In *SaTML*.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2025c. Min-k%++: Improved baseline for pretraining data detection from large language models. In *ICLR*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *arXiv:2205.01068*.

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

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

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv:2307.15043*.

A Related Work

Membership Inference Attacks. Membership inference attacks (MIAs) (Shokri et al., 2017) test whether a specific data sample was included in a model's training set. Traditional MIAs focus on classification models and typically assume access to model confidence scores, loss values, or even internal parameters (Yeom et al., 2018; Salem et al., 2019; Sablayrolles et al., 2019; Leino and Fredrikson, 2020; Nasr et al., 2019; Ye et al., 2022). Label-only MIAs further relax assumptions, requiring only the predicted label (Choquette-Choo et al., 2021; Li and Zhang, 2021; Wu et al., 2024).

Membership Inference Attacks against Pre-trained LLMs. With the widespread adoption of LLMs, privacy risks stemming from the memorization of pre-training data in generative models have drawn increasing attention. Unlike traditional classifiers, LLMs output free-form text, introducing additional challenges for MIAs. Recent studies have demonstrated the feasibility of MIAs against various types of pre-trained LLMs by leveraging token-level logits or probability information (Carlini et al., 2021, 2022; Fu et al., 2025; Mireshghallah et al., 2022; Shi et al., 2024; Duan et al., 2024; Meeus et al., 2024a; Zhang et al., 2024; Kim et al., 2024; Mozaffari and Marathe, 2024; Panda et al., 2025; Huang et al., 2025; Zhang et al., 2025a; Xie et al., 2024; Zhang et al., 2025c; Wang et al., 2025) and even LLMs' parameters and their intermediate results (Li et al., 2023; He et al., 2025a; Tang et al., 2025; Makhija et al., 2025).

In the challenging label-only setting, an adversary can only access the final generated text from a target model, without any exposure to its internal logits or token probabilities. Among existing label-only MIAs for LLMs, some approaches rely on supervised learning, which requires a dedicated training set of labeled member and non-member samples to train an attack model (Zhou et al., 2024; Hu et al., 2025). In contrast, our method operates in an unsupervised manner, obviating the need for such labeled data. The most closely related work to ours is PETAL (He et al., 2025b), which leverages a surrogate language model to approximate the target model's token probabilities. However, despite its novelty, its reliance on a surrogate model makes the approach highly sensitive to surrogate selection and introduces non-negligible prediction errors, given the inevitable gap between any surrogate and the target model.

Other Attacks against LLMs. Beyond membership inference, LLMs are vulnerable to a variety of other attacks that expose different aspects of their security and privacy risks. Jailbreak attacks craft adversarial prompts to bypass safety alignment and elicit harmful or restricted content from aligned models (Zou et al., 2023; Geng et al., 2025). Harmful fine-tuning attacks (Qi et al., 2024; Yi et al., 2025b,d,c) exploit the fine-tuning interface to erase safety guardrails, enabling the resulting model to produce unsafe outputs. Backdoor attacks implant hidden triggers during training or fine-tuning, causing the model to behave normally on clean inputs but produce attacker-specified outputs when the trigger is present (Li et al., 2024; Yi et al., 2024, 2025a).

B On the Prevalence and Practicality of the Label-Only Threat Model

In this section, we provide a detailed justification for our focus on the "label-only" threat model, a setting where an adversary can only access the final generated text from a target model without any accompanying probabilities or logits. We argue that this is not a niche or purely theoretical constraint, but rather reflects the most common and practical scenario for interacting with LLMs in the real world. Our justification is threefold: the prevalence of purely text-based systems, the "illusion of access" in APIs that provide partial logits, and the resulting status of the label-only setting as a universal and foundational challenge.

The Ubiquity of Pure Label-Only Systems. A vast number of real-world LLM applications and services operate in a purely label-only mode. The most prominent examples are the public web interfaces of leading models such as OpenAI's ChatGPT, Google's Gemini, Anthropic's Claude.

In these common scenarios, a user provides a prompt and receives a textual response. The underlying token probabilities, which are essential for conventional logit-based MIAs, are completely hidden. For any adversary interacting with these widespread services, the attack surface is strictly limited to observing input-output text pairs. Therefore, developing robust MIAs for this setting is crucial for auditing the privacy of a significant portion of the deployed LLM ecosystem.

The "Illusion of Access" in Logit-Providing APIs. Even in more advanced APIs from providers like OpenAI that seem to offer richer access via pa-

Table 4: TPR at low FPR (FPR=5%) results on WikiMIA with bold indicating best label-only attack.

		Logits-based Attacks							*Label-only Attacks*	
Len	**Model**	**PPL**	**Ref**	**Zlib**	**Neighbor**	**Mink**	**Mink++**	**RECALL**	**PETAL**	**SEAD**
32	**Mamba-1.4B**	0.13	0.06	0.17	0.06	0.13	0.13	0.46	**0.14**	0.12
	Opt-6.7B	0.11	0.05	0.13	0.06	0.18	0.12	0.24	**0.10**	0.07
	Pythia-6.9B	0.12	0.05	0.16	0.07	0.18	0.15	0.52	**0.16**	0.15
	Llama-13B	0.11	0.10	0.10	0.08	0.19	0.33	0.43	0.09	**0.36**
	Neox-20B	0.20	0.02	0.19	0.14	0.28	0.21	0.48	**0.20**	**0.20**
	Llama-30B	0.18	0.09	0.13	0.08	0.19	0.26	0.51	0.11	**0.27**
64	**Mamba-1.4B**	0.10	0.12	0.14	0.09	0.16	0.14	0.29	0.13	**0.14**
	Opt-6.7B	0.10	0.07	0.12	0.11	0.15	0.17	0.18	0.14	**0.16**
	Pythia-6.9B	0.13	0.05	0.16	0.11	0.19	0.15	0.52	**0.16**	0.15
	Llama-13B	0.11	0.14	0.13	0.10	0.17	0.31	0.66	0.09	**0.30**
	Neox-20B	0.13	0.02	0.19	0.13	0.19	0.22	0.40	0.16	**0.21**
	Llama-30B	0.14	0.12	0.15	0.10	0.17	0.34	0.59	0.11	**0.35**
128	**Mamba-1.4B**	0.12	0.06	0.19	0.16	0.09	0.10	0.37	**0.17**	0.14
	Opt-6.7B	0.13	0.16	0.10	0.13	0.17	0.22	0.24	**0.13**	0.12
	Pythia-6.9B	0.14	0.10	0.12	0.11	0.19	0.20	0.27	**0.25**	0.17
	Llama-13B	0.22	0.17	0.19	0.13	0.20	0.38	0.43	0.14	**0.24**
	Neox-20B	0.18	0.06	0.22	0.15	0.22	0.25	0.33	0.17	**0.24**
	Llama-30B	0.24	0.14	0.18	0.14	0.22	0.22	0.39	0.17	**0.24**
	Average	0.14	0.09	0.15	0.11	0.18	0.22	0.40	0.15	**0.20**

rameters like `logprobs`, the provided information is severely restricted. These APIs almost never return the full probability distribution over the model's entire vocabulary. Instead, they typically provide a "top-k" view of the distribution. For instance, OpenAI's API caps the `top_logprobs` parameter at a maximum of 5, meaning an adversary can only see the probabilities of the five most likely next tokens. This limitation has profound implications for logit-based MIAs. Methods such as Min-K% (Shi et al., 2024) and its variants, which derive their signal from the least likely tokens, are rendered completely ineffective as these crucial low-probability tokens are never exposed. For the vast majority of the vocabulary at any given step, the adversary has zero probability information. This forces any robust MIA to operate in what is, for all practical purposes, a label-only environment.

Moreover, *other major providers, such as Anthropic, do not expose token probabilities at all in their public APIs*, further cementing the prevalence and practical importance of the label-only setting.

Label-Only as a Foundational and Universal Setting. Given the diverse and evolving landscape of API restrictions, the label-only approach stands out as the most universally applicable solution. The one constant across all LLM services, from closed-source commercial APIs to open-source models deployed with simplified interfaces, is their ability to generate text. An attack that relies solely on this fundamental output will remain feasible and relevant, regardless of whether a provider decides to introduce, modify, restrict, or completely remove access to logits in the future.

C Additional Experimental Results

C.1 Additional Main Results

To further evaluate the practical efficacy of SEAD, particularly in scenarios where minimizing false positives is crucial, we present the True Positive Rate (TPR) at a low False Positive Rate (FPR) of 5%. These results, detailed in Table 4 for the WikiMIA benchmark and Table 5 for the MIMIR benchmark, complement the AUROC findings by highlighting attack performance in a stringent, precision-focused regime.

On the WikiMIA benchmark, as shown in Table 4, SEAD achieves an average TPR@5%FPR of 0.20. This is higher than the average of 0.15 achieved by PETAL, the existing label-only baseline. The difference is noticeable on certain larger models; for instance, against Llama-13B (length 32), SEAD's TPR is 0.36 compared to PETAL's 0.09, and against Llama-30B (length 64), SEAD reaches 0.35 versus PETAL's 0.11. Compared with several logits-based methods, SEAD remains competitive at low FPR. It records a higher average TPR than PPL (0.14), Ref (0.09), Zlib (0.16), Neighbor (0.11), and Mink (0.18). Its performance is comparable to Mink++ (0.21), while RECALL (0.40) shows a higher TPR.

Turning to the more challenging MIMIR benchmark (Table 5), SEAD demonstrates continued utility under low-FPR conditions. It achieves an average TPR@5%FPR of 0.058 across all models and sub-datasets, which is higher than PETAL's average of 0.055. Compared to logits-based attacks on MIMIR, SEAD's average TPR@5%FPR (0.058) is comparable to PPL (0.055) and Zlib (0.058), and

Table 5: TPR at low FPR(FPR=5%) results on MIMIR with bold indicating best label-only attack.

	Wikipedia				GitHub				Pile CC				PubMed Central			
Method	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B
Logits-based Attacks:																
PPL	0.06	0.05	0.06	0.06	0.10	0.05	0.05	0.06	0.01	0.02	0.02	0.03	0.04	0.04	0.04	0.03
Ref	0.03	0.04	0.04	0.04	0.07	0.09	0.08	0.09	0.04	0.03	0.04	0.03	0.03	0.02	0.02	0.04
Zlib	0.06	0.06	0.06	0.07	0.10	0.06	0.06	0.09	0.02	0.02	0.03	0.03	0.04	0.04	0.03	0.02
Neighbor	0.07	0.08	0.05	0.08	0.09	0.08	0.08	0.09	0.04	0.04	0.05	0.06	0.03	0.04	0.07	0.04
Mink	0.07	0.08	0.07	0.08	0.10	0.05	0.06	0.07	0.04	0.06	0.04	0.04	0.03	0.04	0.04	0.03
Mink++	0.08	0.09	0.09	0.10	0.08	0.04	0.06	0.06	0.06	0.05	0.06	0.08	0.06	0.04	0.05	0.05
RECALL	0.08	0.08	0.07	0.09	0.16	0.18	0.14	0.15	0.06	0.04	0.05	0.06	0.02	0.02	0.04	0.02
Label-only Attacks:																
PETAL	**0.09**	0.05	**0.06**	0.05	**0.08**	**0.12**	0.07	0.08	0.02	0.03	**0.04**	0.02	0.04	**0.04**	0.02	0.02
SEAD	0.05	**0.07**	0.05	**0.09**	0.07	0.10	**0.16**	**0.10**	**0.04**	**0.04**	0.03	**0.04**	**0.05**	**0.04**	**0.03**	**0.03**

	ArXiv				DM Mathematics				HackerNews				Average			
Method	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B	1.4B	2.8B	6.9B	12B
Logits-based Attacks:																
PPL	0.06	0.07	0.07	0.06	0.04	0.01	0.04	0.05	0.10	0.07	0.11	0.11	0.06	0.04	0.06	0.06
Ref	0.05	0.06	0.07	0.06	0.06	0.05	0.04	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Zlib	0.05	0.05	0.05	0.05	0.05	0.03	0.06	0.04	0.10	0.11	0.11	0.11	0.06	0.05	0.06	0.06
Neighbor	0.04	0.04	0.05	0.05	0.07	0.06	0.07	0.05	0.08	0.09	0.11	0.10	0.06	0.06	0.07	0.07
Mink	0.04	0.06	0.07	0.06	0.05	0.04	0.03	0.04	0.12	0.12	0.10	0.12	0.07	0.06	0.06	0.07
Mink++	0.06	0.05	0.04	0.06	0.05	0.05	0.04	0.06	0.07	0.06	0.09	0.11	0.06	0.06	0.06	0.07
RECALL	0.08	0.06	0.06	0.06	0.06	0.04	0.10	0.08	0.06	0.09	0.08	0.08	0.08	0.07	0.08	0.08
Label-only Attacks:																
PETAL	**0.05**	**0.06**	**0.06**	0.05	**0.03**	**0.04**	**0.04**	**0.04**	**0.09**	0.10	0.09	**0.10**	**0.06**	**0.06**	0.05	0.05
SEAD	0.04	**0.06**	0.05	**0.06**	0.02	0.03	0.02	**0.04**	**0.09**	**0.12**	**0.10**	0.08	0.05	**0.06**	**0.06**	**0.06**

higher than Ref (0.043). It is slightly lower than Neighbor (0.065), Mink (0.065), Mink++ (0.063), and RECALL (0.078).

Collectively, these TPR@5%FPR results indicate SEAD's capability to identify member samples with greater precision than the existing label-only method, PETAL, especially when false alarms must be strictly controlled. Its standing relative to several logits-based attacks in this practical setting suggests its utility as an MIA tool in label-only scenarios.

C.2 SEAD as a Foundational Density Estimator for Building More Sophisticated Label-Only Attacks

Table 6: AUROC Comparison between Standalone SEAD and SEAD+RECALL.

Methods	Mamba-1.4B	OPT-6.7B	Pythia-6.9B
PETAL	0.60	0.60	0.62
SEAD	0.63	0.62	0.66
PETAL+RECALL	0.67	0.60	0.70
SEAD+RECALL	**0.75**	**0.63**	**0.80**

A key advantage of SEAD is its ability to provide robust density estimations in the label-only setting, making it a versatile building block for more sophisticated attacks. To demonstrate this, we investigate whether advanced principles from logits-based methods can be adapted to the label-only setting using SEAD as the core engine.

We focus on the insightful calibration strategy of RECALL (Xie et al., 2024), a state-of-the-art logits-based MIA. RECALL's core logic is to measure the relative change in model confidence (typically log-likelihood) when a non-member prefix is prepended to a sequence. We adapt this strategy to the label-only setting by replacing the unavailable log-likelihood metric with our SEAD's semantic-aware density score. This integrated approach, which we term RECALL+SEAD, creates a novel and more powerful label-only attack.

The AUROC results of this integration are presented in Table 6. The findings clearly show that RECALL+SEAD consistently and significantly outperforms the standalone SEAD. For instance, on Mamba-1.4B, the AUROC dramatically increases from 0.63 (SEAD) to 0.75 (RECALL+SEAD). A similar substantial improvement is observed for Pythia-6.9B, where the AUROC rises from 0.66 to 0.80. Even for OPT-6.7B, where the standalone SEAD performance was more modest, the integration yields an improvement from 0.62 to 0.63.

This performance uplift underscores the effectiveness of combining SEAD's robust label-only density estimation with RECALL's calibration strategy. SEAD provides a reliable way to quantify the model's "belief" or density assignment in the label-only scenario, which then serves as a high-quality input for RECALL's comparative logic. The success of RECALL+SEAD demonstrates that SEAD can serve as a powerful building block, en-

abling the adaptation of sophisticated, proven MIA principles from the logits-based domain to the more challenging but practical label-only setting. This opens promising avenues for developing even more potent label-only MIAs by leveraging SEAD's core density computation capabilities.

C.3 Additional Ablation Study

NLI vs. Cosine Similarity. To justify our choice of using an NLI model for semantic estimation, we conducted an ablation study comparing our method against a variant that uses cosine similarity from a sentence embedding model (`all-MiniLM-L12-v2`). The results in Table 7 show that the NLI-based approach consistently and significantly outperforms its cosine similarity counterpart.

The Influence of Different NLI Models. We utilize an NLI model, `deberta-xlarge-mnli`, as the default NLI model to compute the semantic-aware density in our pipeline. Here, we further investigate the impact of using other NLI models on MIA performance. We adopt another two NLI models *mDeBERTa-v3-base-xnli-multilingual-nli-2mil7* and *xlm-v-base-mnli-xnli*. The experimental results, as shown in Table 8, indicate that SEAD shows consistently good performance when integrated with different NLI models.

C.4 Versatility of the Semantic-Aware Density Module with Ground-Truth Logits

In the main body of our work, we introduced the semantic-aware density module as a crucial component to enhance the performance of lexical density estimated via Monte Carlo sampling in a label-only setting. A natural and important question arises: is the benefit of our semantic-aware module limited only to MC-based probability estimates, or is it a more fundamental and general-purpose enhancement for MIA?

To investigate this, we conducted an ablation study to assess whether the semantic-aware module could also improve the performance of lexical density calculated directly from the model's ground-truth logits. This experiment aims to demonstrate the module's versatility and its ability to function as a powerful, standalone enhancement for traditional logit-based attacks, such as PPL.

Experimental Setup. For this experiment, we utilized the WikiMIA benchmark. Our baseline is the PPL attack, which uses the standard lexical density calculated using the ground-truth logits provided by the model. For our proposed method, we applied the semantic-aware density logic to these ground-truth probabilities. Due to the computational infeasibility of running NLI comparisons against the entire vocabulary (>50k tokens), we constrained our semantic-aware calculation to the **Top-2000** most probable tokens for each position.

Results and Analysis. The results, presented in Table 9, clearly demonstrate that the semantic-aware module provides a consistent and significant performance uplift across all tested models, even when applied to ground-truth logits.

As demonstrated in Table 9, the AUROC for `Mamba-1.4B` improved from 0.60 to 0.67, and for `Pythia-6.9B` it increased from 0.62 to 0.68. This improvement underscores that the core limitation of lexical density, its reliance on exact token matches, is a fundamental issue that exists regardless of whether the probabilities are estimated via MC sampling or accessed directly. By aggregating probability mass from semantically equivalent tokens (such as considering "America" when the ground truth is "USA"), our module captures a more robust and accurate signal of the model's memorization, which is often expressed through conceptual understanding rather than verbatim repetition.

This experiment validates the versatility and general applicability of our semantic-aware density module. It positions the module not just as a component for our label-only attack, but as a powerful, standalone technique for enhancing a broad class of MIA methods that rely on token probabilities.

C.5 Results on WikiMIA When the Input Text is Paraphrased

Text paraphrasing is a common data augmentation technique in NLP and can also be viewed as a potential defense against MIAs. Since *paraphrased members* are semantically highly similar to the original ones and can also reveal their existence, the current community generally agrees that MIAs against LLMs should infer *paraphrased members* as members (Shi et al., 2024; Zhang et al., 2025c). To investigate the robustness of SEAD against such semantic alterations, we evaluate its performance on the paraphrased WikiMIA benchmark provided by Zhang et al. (2025c), comparing it against various logits-based attacks and PETAL.

The AUROC results, presented in Table 10, indicate that SEAD maintains robust performance even when the input text is paraphrased. SEAD achieves an average AUROC of 0.71 on the paraphrased dataset. *Notably, this represents the same*

Table 7: AUROC comparison of SEAD using NLI-based entailment versus cosine similarity.

Semantic Estimator	Mamba-1.4B	OPT-6.7B	Pythia-6.9B
SEAD-CS (Cosine Similarity)	0.59	0.59	0.60
SEAD-NLI (Ours)	**0.65**	**0.63**	**0.66**

Table 8: AUROC comparison of different NLI models.

NLI Models	Mamba-1.4B	OPT-6.7B	Pythia-6.9B
deberta-xlarge-mnli	0.63	0.63	0.66
mDeBERTa-v3-base-xnli-multilingual-nli-2mil7	0.66	0.63	0.65
xlm-v-base-mnli-xnli	0.63	0.63	0.66

Table 9: AUROC comparison on the WikiMIA benchmark. Integrating our semantic-aware module with ground-truth logits (PPL + Semantic-Aware) consistently outperforms the standard PPL attack.

Model	PPL	PPL + Semantic-Aware
Mamba-1.4B	0.60	**0.67**
Opt-6.7B	0.59	**0.64**
Pythia-6.9B	0.62	**0.68**

performance as its average AUROC of 0.71 on the original (non-paraphrased) WikiMIA benchmark (as detailed in Table 1 for SEAD), underscoring SEAD's strong resilience to the semantic variations introduced by paraphrasing.

This resilience may be attributed to the semantic-aware density component, which is designed to capture such semantic equivalences. By aggregating probability mass from tokens that are semantically similar (as determined by the NLI model) to the ground truth, SEAD can effectively recognize paraphrased members even if the exact lexical tokens differ from the original training instance.

C.6 Efficiency Analysis

Table 12 illustrates the inference speed and AUROC of different label-only MIA methods against different pre-trained LLMs on the WikiMIA dataset. We query the MIA methods with 50 samples (i.e., 50 distinct text inputs for membership inference testing) and record the average inference speed (sec/sample), defined as the average time consumption for processing a single sample. For methods like SEAD that involve internal Monte Carlo sampling, the number of internal samples per token is specified in the table.

As observed in Table 12, our default SEAD configuration, utilizing 50 Monte Carlo samples per token (SEAD sample nums=50), achieves the highest average AUROC of 0.65. This represents a notable improvement over PETAL (0.61 AUROC). However, this enhanced performance comes at the cost of increased inference time, with SEAD (sample nums=50) averaging 6.47 sec/sample compared to PETAL's 5.40 sec/sample.

To investigate a more computationally efficient variant, we also evaluated SEAD with a reduced number of Monte Carlo samples, specifically 10 samples per token (SEAD sample nums=10). This configuration exhibits an average inference speed of 3.61 sec/sample, which is considerably faster than PETAL's speed. Notably, even with this significant reduction in sampling numbers, SEAD (sample nums=10) still achieves an average AUROC of 0.61. This result matches PETAL's performance while operating more efficiently, demonstrating that **SEAD offers a superior accuracy-efficiency trade-off**. This flexibility allows practitioners to choose a SEAD configuration that balances desired attack efficacy with available budget.

The practicality of this approach is greatly enhanced by using batch generation features of existing LLMs (e.g., `num_return_sequences`). When requesting K samples in a single API call, this computationally costly context processing is performed only once. This is substantially faster than making K separate API calls, each of which would require re-processing the entire context from scratch. For instance, generating 50 tokens from OPT-6.7B via 50 individual queries takes an average of 5.26s, whereas a single batched query with `num_return_sequences=50` achieves the same result in just 0.12s.

C.7 Case Study on a Representative Closed-source Commercial LLM

Evaluating MIAs against closed-source LLMs is inherently challenging due to the non-disclosure of their pre-training data. This lack of transparency makes it difficult to establish a strictly controlled experimental environment that perfectly mirrors the model's training conditions. To navigate this,

Table 10: AUROC on WikiMIA when the input text is paraphrased with bold indicating best label-only attack.

	Logits-based Attacks							Label-only Attacks	
Model	**PPL**	**Ref**	**Zlib**	**Neighbor**	**Mink**	**Mink++**	**RECALL**	**PETAL**	**SEAD**
Mamba-1.4B	0.60	0.60	0.61	0.58	0.62	0.67	0.92	0.60	**0.64**
Opt-6.7B	0.60	0.45	0.61	0.61	0.60	0.64	0.84	0.61	**0.63**
Pythia-6.9B	0.64	0.44	0.64	0.63	0.65	0.67	0.82	0.64	**0.67**
Llama-13B	0.68	0.60	0.68	0.64	0.66	0.83	0.91	0.62	**0.79**
Neox-20B	0.69	0.53	0.69	0.65	0.70	0.71	0.86	0.67	**0.71**
Llama-30B	0.70	0.60	0.70	0.65	0.69	0.81	0.92	0.64	**0.82**
Average	0.65	0.54	0.66	0.63	0.65	0.72	0.88	0.63	**0.71**

Table 11: TPR@5%FPR on WikiMIA when the input text is paraphrased with bold indicating best label-only attack.

	Logits-based Attacks							Label-only Attacks	
Model	**PPL**	**Ref**	**Zlib**	**Neighbor**	**Mink**	**Mink++**	**RECALL**	**PETAL**	**SEAD**
Mamba-1.4B	0.15	0.07	0.13	0.09	0.11	0.09	0.49	**0.13**	0.12
Opt-6.7B	0.11	0.03	0.12	0.12	0.12	0.07	0.27	0.10	**0.11**
Pythia-6.9B	0.15	0.04	0.13	0.14	0.22	0.15	0.37	**0.16**	0.14
Llama-13B	0.16	0.12	0.15	0.17	0.14	0.33	0.44	0.14	**0.35**
Neox-20B	0.18	0.02	0.19	0.21	0.19	0.13	0.38	**0.18**	0.15
Llama-30B	0.15	0.13	0.15	0.13	0.18	0.27	0.52	0.13	**0.29**
Average	0.15	0.07	0.15	0.14	0.16	0.17	0.41	0.14	**0.19**

our evaluation on GPT-3.5-Turbo-Instruct largely follows the setup outlined in the PETAL study (He et al., 2025b), utilizing an updated version of the WikiMIA benchmark (Fu et al., 2025).

SEAD demonstrates superior performance to PETAL against the commercial LLM GPT-3.5-Turbo-Instruct. The attack results on GPT-3.5-Turbo-Instruct, presented in Table 13, illustrate this advantage. SEAD achieved an AUROC of 0.63, compared to PETAL's 0.57. Furthermore, in terms of TPR@5%FPR, SEAD obtained 11.22%, significantly higher than PETAL's 7.14%. These results indicate that SEAD maintains its superior performance compared to PETAL when evaluated on the commercial model in label-only settings.

D Additional Baseline Details

For a target data point $\mathbf{x}$, a membership inference attack (MIA) evaluates whether $\mathbf{x}$ was included in the training dataset D of a model M by computing a membership score $S(\mathbf{x}; M)$. Below, we detail the baseline MIA methods employed in our study.

D.1 Perplexity (PPL) Attack

The PPL baseline (Carlini et al., 2021; Yeom et al., 2018) computes the exponentiated cross-entropy loss of the target sequence under the model, where lower perplexity suggests higher likelihood of membership. The membership score is derived as the negative perplexity to align with the convention that higher scores indicate membership.

$$S(\mathbf{x}; M) = -\text{PPL}(x) \tag{8}$$
$$= -\exp\left(-\frac{1}{n}\sum_{i=1}^{n}\log p(t_i|t_{<i})\right)$$

D.2 Reference Attack

Reference attack baseline (Carlini et al., 2021) adjusts the target model's PPL using a reference model M_{ref} trained on a similar but distinct dataset. This calibration accounts for sample-specific complexity, reducing false positives. When a reference model is required, we follow Shi et al. (2024) and use a smaller model from the same family (Pythia-70M for Pythia models). For other cases, we set it to GPT2-XL.

$$S(\mathbf{x}; M) = \text{PPL}(\mathbf{x}; M_{\text{ref}}) - \text{PPL}(\mathbf{x}; M) \tag{9}$$

D.3 Zlib Entropy

The Zlib attack method (Carlini et al., 2021) scales the model's PPL by the zlib compression size of the input. Member samples are expected to exhibit lower entropy, resulting in smaller compressed sizes relative to their PPL.

$$S(\mathbf{x}; M) = -\frac{\text{PPL}(\mathbf{x}; M)}{\text{zlib}(\mathbf{x})} \tag{10}$$

D.4 Neighborhood Attack

Neighborhood attack baseline (Mattern et al., 2023) compares the PPL of the target sample to the average PPL of its perturbed variants. Member samples typically show greater resilience to perturbations,

Table 12: Comparison of inference speed (sec/sample) and AUROC across different label-only methods.

Models	PETAL		SEAD (sample nums=10)		SEAD (sample nums=50)	
	Inference speed	AUROC	Inference speed	AUROC	Inference speed	AUROC
Mamba-1.4B	9.52	0.60	7.77	0.60	10.83	0.65
OPT-6.7B	3.25	0.60	1.58	0.59	3.56	0.62
Pythia-6.9B	3.43	0.62	1.49	0.63	5.03	0.67
Average	5.40	0.61	3.61	0.61	6.47	0.65

Table 13: Attack results on GPT-3.5-Turbo-Instruct. The dataset is an updated version of WikiMIA.

Methods	PETAL		SEAD	
Metrics	AUROC	TPR@5%FPR	AUROC	TPR@5%FPR
GPT-3.5	0.57	7.14%	0.63	11.22%

leading to larger loss discrepancies. For evaluating the neighborhood attack, we use 10 neighbor samples for every text input to calculate its intrinsic hardness. As synthetically generating perturbed versions of text input can be time-consuming, we use perturbed WikiMIA[2] by (Zhang et al., 2025c) and perturbed MIMIR[3] by (Duan et al., 2024). The score is calculated using the following formula, where k is the number of neighbors and $\tilde{\mathbf{x}}_i$ is the i-th perturbed variant of the original sample $\mathbf{x}$:

$$S(\mathbf{x}; M) = \frac{1}{k}\sum_{i=1}^{k} \text{PPL}(\tilde{\mathbf{x}}_i; M) - \text{PPL}(\mathbf{x}; M) \tag{11}$$

D.5 Min-K% Probability Attack

The Min-K% method (Shi et al., 2024) computes membership scores using the average negative log-likelihood of the least probable $k\%$ tokens. Member samples are hypothesized to assign higher probabilities to these tokens than non-members. For evaluating MIN-K% PROB, we set k to 20 as done in the original paper.

$$S(\mathbf{x}; M) = \frac{1}{|\text{min-}k(\mathbf{x})|} \sum_{x_i \in \text{min-}k(\mathbf{x})} -\log p(x_i|x_{<i}) \tag{12}$$

D.6 Min-K%++ Attack

The Min-K%++ method is an enhanced version of Min-K% (Zhang et al., 2025c), this method normalizes token-level log-likelihoods using the mean ($\mu_{x_{<t}}$) and standard deviation ($\sigma_{x_{<t}}$) of the model's vocabulary distribution for the prefix $x_{<t}$. For evaluating Min-K%++, we also set $k = 20$.

$$S_{\text{token}}(x_{<t}, x_t; M) = \frac{\log p(x_t|x_{<t}; M) - \mu_{x_{<t}}}{\sigma_{x_{<t}}}, \tag{13}$$

$$S(\mathbf{x}; M) = \frac{1}{|\text{min-}k\%|} \sum_{\substack{(x_{<t}, x_t) \\ \in \text{min-}k\%}} S_{\text{token}}(x_{<t}, x_t; M). \tag{14}$$

D.7 RECALL Attack

The RECALL attack (Xie et al., 2024) baseline leverages a known non-member prefix P and evaluates how the model's confidence in generating a target sequence $\mathbf{x}$ changes when conditioned on P. For each input, the model computes the log-likelihood (LL) of the target sequence both with and without the prefix. Since LL values are negative, the RECALL score is greater than 1 if conditioning on P leads to a decrease in confidence.

$$S(\mathbf{x}; M) = \frac{LL(\mathbf{x}|P)}{LL(\mathbf{x})} \tag{15}$$

D.8 PETAL Attack

The PETAL baseline (He et al., 2025b) approximates the perplexity of a sequence without accessing the model's internal probabilities, by leveraging the semantic similarity between sampled tokens and the ground truth tokens. Given a sequence $\mathbf{x} = t_1, \ldots, t_n$, PETAL first samples a token $\hat{t}_i$ from the model for each position i, and measures the similarity $\text{sim}(t_i, \hat{t}i)$. We utilize GPT2-XL (Radford et al., 2019) as the surrogate model as done in the original paper.

$$S_{token}(\mathbf{x}; M) = \beta \cdot \text{sim}(t_i, \hat{t}_i) + \alpha \tag{16}$$

$$S(\mathbf{x}; M) = -\frac{1}{n}\sum_{i=1}^{n} S(\mathbf{x}; M) \tag{17}$$

The coefficients α and β are obtained by fitting a linear regression between similarity scores and actual log-probabilities using a surrogate model with known access.

[2] https://huggingface.co/datasets/zjysteven/WikiMIA_paraphrased_perturbed

[3] https://huggingface.co/datasets/iamgroot42/mimir

E Reproducibility Statement

The detailed experimental settings of datasets, models, hyper-parameter settings, and computational resources can be found in Section 4.1 and Section D. The codes and model checkpoints for reproducing our main evaluation results are provided in the anonymous GitHub repository.

F Discussion on Adopted Data

In our experiments, we employ open-source benchmark datasets to evaluate the effectiveness of SEAD. All datasets used in this work are publicly released research artifacts, designed specifically for auditing and evaluating the privacy properties of pre-trained language models. We strictly adhere to their open-source licenses and utilize them solely for research purposes.

Dataset Sources and Splits. We adopt two widely recognized benchmarks: WikiMIA (Shi et al., 2024) and MIMIR (Duan et al., 2024). Both datasets follow standardized official splits. For instance, in our experiments on WikiMIA with a sequence length of 32, a total of 776 samples are used, maintaining a balanced 1:1 ratio between member and non-member instances.

Compliance and Intended Use. Our research strictly adheres to the open-source licenses of these datasets. The usage of WikiMIA and MIMIR in SEAD is fully consistent with their intended purposes, serving as benchmarks for evaluating membership inference attacks against pre-trained LLMs.

Privacy and Sensitivity. This study does not involve any private, non-public, or sensitive user information. All datasets employed are constructed from publicly available text corpora such as Wikipedia, curated for scientific evaluation. Therefore, our experiments raise no privacy, ethical, or data protection concerns.