MACHINE TEXT DETECTORS ARE MEMBERSHIP INFERENCE ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Although membership inference attacks (MIAs) and machine-generated text detection target different goals, their methods often exploit similar signals based on a language model's probability distribution, and the two tasks have been studied independently. This can result in conclusions that overlook stronger methods and valuable insights from the other task. In this work, we theoretically and empirically demonstrate the *transferability*, i.e., how well a method originally developed for one task performs on the other, between MIAs and machine text detection. We prove that the metric achieving asymptotically optimal performance is identical for both tasks. We unify existing methods under this optimal metric and hypothesize that the accuracy with which a method approximates this metric is directly correlated with its transferability. Our large-scale empirical experiments demonstrate a strong rank correlation ($\rho \approx 0.7$) in cross-task performance. Notably, we also find that a detector achieves the strongest performance among evaluated methods on both tasks, demonstrating the practical impact of transferability. To facilitate cross-task development and fair evaluation, we introduce MINT, a unified evaluation suite for MIAs and detection, implementing 15 recent methods from both tasks.[1]

1 INTRODUCTION

Large language models (LLMs) have demonstrated human-level generative and understanding capabilities, impacting fields such as creative writing (Chakrabarty et al., 2024), news reporting (Futurism, 2023), and even scientific discovery (Lu et al., 2024). Despite many positive societal implications, their negative consequences have increasingly been reported. For instance, LLMs may leak personal (Lukas et al., 2023) or copyrighted information (Wei et al., 2024) due to their memorization of training data (Carlini et al., 2021; Morris et al., 2025). In addition to privacy issues, LLMs also raise challenges to authorship authenticity, as they can be exploited for mass-producing propaganda (Goldstein et al., 2024) or cheating on student assignments (Guardian, 2025).

Many recent works aim to mitigate such negative implications of LLMs. One major direction is membership inference attacks (MIAs), which aim to classify whether a given text sample is a member of a language model's training data (Carlini et al., 2022; Mattern et al., 2023; Shi et al., 2024). This helps identify potential leaks of personal information and copyright infringement. To address the challenges of authorship authenticity, another major line of work is machine-generated text detection, which distinguishes between human-written and machine-generated texts (Ippolito et al., 2020; Mitchell et al., 2023; Hans et al., 2024; Yang et al., 2024). This safeguards against misuse of LLMs (e.g., misinformation and academic misconduct) by flagging suspicious texts.

While these two tasks target different goals, their methods often leverage similar signals based on a language model's probability distribution. In MIAs, documents that are members of a model's training dataset tend to show higher likelihoods under the model than non-members. Similarly, in machine text detection, machine-generated texts typically exhibit higher likelihoods under the target model than human-written texts. Thus, both tasks use text likelihood or entropy as standard baselines. Indeed, as illustrated in Figure 1, our pilot study shows strikingly similar score distributions of the state-of-the-art MIA and detector across both tasks. Despite this shared property, the two tasks have been studied independently. This can result in biased evaluations or conclusions that overlook stronger methods and valuable insights from the other task.

[1]Code and datasets are available at `https://anonymous.4open.science/r/mint-B44A/`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

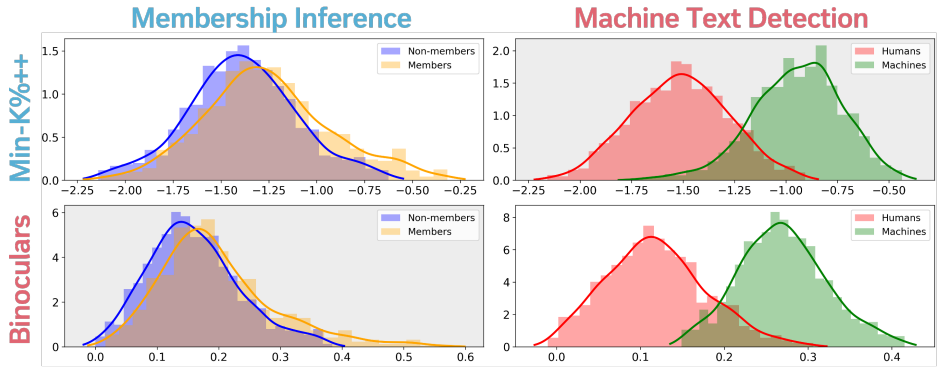

Figure 1: Predicted score distributions of *Min-K%++* (state-of-the-art MIA) and *Binoculars* (state-of-the-art detector) across both tasks. Shaded areas indicate the cross-task setting. Although both methods were developed for two separate tasks, the distributions across populations induced by both methods are **strikingly similar**, suggesting their transferability. See §3.3 for more details.

Motivated by this gap, we theoretically and empirically study the *transferability*, i.e., how well a method originally developed for one task performs on the other, between MIAs and machine text detection. For our theoretical contribution, we prove that both tasks share the same optimal metric for achieving asymptotically highest performance: *the likelihood ratio test between the target model distribution and the true population distribution* (§2.2). We unify a large proportion of the existing methods from both tasks as approximations of this optimal metric and hypothesize that the transferability of a method is correlated with how well it approximates this metric.

To empirically quantify the transferability, we conduct large-scale experiments, spanning 7 state-of-the-art MIA methods and 5 state-of-the-art machine text detectors across 13 domains and 10 generators (§3.1). For each method, we evaluate its performance on both tasks, MIAs and machine text detection, and then compute the rank correlation between their performance rankings on the two tasks. We find that many methods effective in MIAs remain effective in machine text detection, and vice versa, with a strong rank correlation ($\rho \approx 0.7$) in their cross-task performance (§3.2). Furthermore, we notably find that *Binoculars* (Hans et al., 2024), originally designed for machine text detection, achieves the strongest performance among evaluated methods on both tasks, demonstrating the practical impact of transferability.

Contributions. Our contributions are as follows: (1) We are the first to provide proof that MIAs and machine text detection share the same asymptotically optimal metric. This identifies what determines performance in both tasks, providing practical guidance for developing stronger methods. (2) We show that many popular methods can be viewed as approximations to this shared optimal metric, which explains when and why transferability arises and how its strength depends on the degree of this approximation. (3) We present the first comprehensive empirical study of transferability between the two tasks, showing a strong rank correlation in cross-task performance. We further find that a machine text detector achieves the strongest performance among evaluated methods on both tasks, highlighting the need for fair and unified evaluation.

2 Theoretical Transferability between MIAs and MGTD

2.1 Task Formulation

Let $\mathcal{X}$ denote the set of all token sequences. Let $P_{\mathcal{Q}}$ be the "true" probability distribution of human-written text over $\mathcal{X}$. Let $\mathcal{M}$ be a language model that induces a distribution $P_{\mathcal{M}}$ over $\mathcal{X}$.

Machine Text Detection. The objective is to determine whether a given text $x \in \mathcal{X}$ is written by humans or generated by the model $\mathcal{M}$. This can be formalized as the following hypothesis test

$$H_0 : x \sim P_{\mathcal{Q}}, \quad H_1 : x \sim P_{\mathcal{M}}.$$

Membership Inference. The objective is to determine whether or not the model $\mathcal{M}$ was trained on a given text $x \in \mathcal{X}$. Let $\mathcal{T}$ denote a training algorithm and let $\mathcal{D}$ be a random training dataset sampled from $P_{\mathcal{Q}}$. This can be formalized as the following hypothesis test, which differs only in whether x is included during training,

$$H_0 : \mathcal{M} \leftarrow \mathcal{T}(\mathcal{D}), \quad H_1 : \mathcal{M} \leftarrow \mathcal{T}(\mathcal{D} \cup \{x\}).$$

The goal of both tasks is to design a test statistic $f(x; \mathcal{M})$ that rejects the respective null hypothesis H_0 with maximal statistical power.

2.2 Unifying MIAs and Machine Text Detection as Likelihood Ratio Tests

Theorem 2.2 (Unified Optimality). Let $\mathcal{X}$ denote the set of all token sequences. Let $\mathcal{M}$ be a language model trained to maximize the likelihood of a training set $\mathcal{D} \subset \mathcal{X}$, and let $P_{\mathcal{M}}$ denote the probability distribution induced by $\mathcal{M}$. Let $P_{\mathcal{Q}}$ denote a probability distribution over the subset of $\mathcal{X}$ that models the "true" human distribution of text. Then the test statistic

$$\Lambda(x) = \frac{P_{\mathcal{M}}(x)}{P_{\mathcal{Q}}(x)}$$

achieves optimal accuracy at a given false positive rate (Type I error) *for both machine text detection and membership inference* under standard asymptotic regularity conditions, with corresponding maximum advantage (improvement over random guessing) bounded by

$$\text{adv} \leq \sqrt{\frac{D_{\text{KL}}(P_{\mathcal{Q}} \| P_{\mathcal{M}})}{8}}.$$

Proof. We prove this by showing that $\Lambda(x)$ coincides with the likelihood ratio test for both tasks.

Step 1: Machine text detection. Consider testing

$$H_0 : x \sim P_{\mathcal{Q}}, \quad H_1 : x \sim P_{\mathcal{M}}.$$

The likelihood ratio test for this hypothesis is

$$\Lambda_{\text{MGT}}(x) = \frac{P_{\mathcal{M}}(x)}{P_{\mathcal{Q}}(x)}$$

which exactly coincides with the proposed statistic $\Lambda(x)$. By the Neyman-Pearson lemma (Neyman & Pearson, 1933), this test statistic is most powerful at a given Type I error.

Step 2: Membership inference. Consider testing

$$H_0 : \mathcal{M} \leftarrow \mathcal{T}(\mathcal{D}), \quad H_1 : \mathcal{M} \leftarrow \mathcal{T}(\mathcal{D} \cup \{x\}).$$

Following Carlini et al. (2022), membership inference can be formulated as a likelihood ratio test between the distributions over trained models with and without the target data point x,

$$\Lambda_{\text{MI}}(x) = \frac{P(\mathcal{M} \mid H_1)}{P(\mathcal{M} \mid H_0)},$$

where $P(\mathcal{M} \mid H)$ denotes the likelihood evaluated at the observed trained model under the corresponding training hypothesis. The exact likelihood over models is intractable. Under our assumptions, however, the likelihood ratio depends on the trained model only through the scalar statistic

$$s(x; \mathcal{M}) = P_{\mathcal{M}}(x),$$

which is sufficient for distinguishing H_0 and H_1 (see Appendix A for a detailed proof of this). Accordingly, the likelihood ratio can be rewritten as

$$\Lambda_{\text{MI}}(x) = \frac{P(s(x; \mathcal{M}) \mid H_1)}{P(s(x; \mathcal{M}) \mid H_0)}.$$

Under the asymptotic assumption that $\mathcal{M}$ has infinite capacity and perfectly maximizes the likelihood of the training set, $P_{\mathcal{M}}(x)$ equals the empirical frequency of x in the training set $\mathcal{D}$. Let $K = N \cdot s(x; \mathcal{M})$ denote the number of occurrences of x in the training set, where $N = |\mathcal{D}|$. Then the likelihood ratio further reduces to

$$\Lambda_{\text{MI}}(x) = \frac{P(K \mid H_1)}{P(K \mid H_0)}.$$

Let q_x be shorthand for $P_{\mathcal{Q}}(x)$, we have

$$K \mid H_0 \sim \text{Binomial}(N, q_x),$$
$$K \mid H_1 \sim 1 + \text{Binomial}(N-1, q_x).$$

For any $k \in \{1, \ldots, N\}$,

$$\frac{\Pr[K = k \mid H_1]}{\Pr[K = k \mid H_0]} = \frac{\binom{N-1}{k-1} q_x^{k-1}(1-q_x)^{N-k}}{\binom{N}{k} q_x^{k}(1-q_x)^{N-k}} = \frac{k}{N q_x}.$$

Substituting $k = N \cdot p_x$ with $p_x = P_{\mathcal{M}}(x)$ yields

$$\Lambda_{\text{MI}}(x) = \frac{p_x}{q_x}.$$

Therefore, the likelihood ratio test for membership inference reduces to

$$\Lambda_{\text{MI}}(x) = \frac{P_{\mathcal{M}}(x)}{P_{\mathcal{Q}}(x)}.$$

Step 3: Advantage bound. For a binary hypothesis test with statistic $\Lambda(x)$, the Bayes error is

$$\varepsilon^* = \frac{1 - \text{TV}(P_{\mathcal{M}}, P_{\mathcal{Q}})}{2},$$

where $\text{TV}(\cdot, \cdot)$ denotes total variation distance. Applying Pinsker's inequality,

$$\text{TV}(P_{\mathcal{M}}, P_{\mathcal{Q}}) \leq \sqrt{\frac{1}{2} D_{\text{KL}}(P_{\mathcal{Q}} \| P_{\mathcal{M}})},$$

yields the stated bound on the error and the corresponding maximum advantage. □

Remark. When $\mathcal{M}$ has sufficient capacity and is trained through asymptotically many epochs on the training set, the statistic $\Lambda(x)$ on membership inference will approach the optimal bound. In practice, however, language models are typically trained with only a single pass over the data, and other creative approximations of $P(x \in \mathcal{D})$ may outperform $P_{\mathcal{M}}(x)$. Our results do not discourage further work on these approximations, but rather provide a theoretical framework for understanding the fundamental connection between machine text detection and membership inference.

Discussion. A key implication of this result is that a method that effectively approximates $\Lambda(x)$ will have high transferability, i.e., perform well on both machine text detection and membership inference. In the remainder of this paper, we analyze a collection of state-of-the-art methods using this framework and measure their degree of transferability in the context of this result.

2.3 Classifying Methods as Approximate Likelihood Ratio Tests

Since the true population distribution $P_{\mathcal{Q}}$ of human-written text is inaccessible, methods in both tasks employ various strategies to approximate it. They can be broadly divided into two approaches:

Approximation via External Reference. The first approach approximates the true distribution by leveraging an external distribution $P_{\mathcal{M}_{\text{ref}}}$ as a surrogate for the true population distribution $P_{\mathcal{Q}}$,

$$P_{\mathcal{Q}}(x) \approx P_{\mathcal{M}_{\text{ref}}}(x).$$

The surrogate distribution $P_{\mathcal{M}_{\text{ref}}}$ can be approximated via another language model (*Reference* (Carlini et al., 2021) from MIAs), a byte-level frequency distribution induced by Huffman encoding (*Zlib* (Carlini et al., 2021) from MIAs), a token-level frequency distribution from an external corpus (*DC-PDD* (Zhang et al., 2024) from MIAs), or cross-model entropy (*Binoculars* (Hans et al., 2024) from machine text detection).

Table 1: Unified formulations of membership inference and machine text detection methods. $P_{\mathcal{M}}(x)$: text likelihood. $\mathcal{R}(x;\mathcal{M})$: average log rank. $\phi(x)$: an arbitrary perturbation function. $\Phi(x) = x/\sigma_x$: division by the standard deviation. See Appendix B for more details on how we derive each formula.

Method	Task	Equational form
Reference (Carlini et al., 2021)	MIA	$-\log(P_{\mathcal{M}}(x)/P_{\mathcal{M}_{\text{ref}}}(x))$
Zlib (Carlini et al., 2021)	MIA	$-\log P_{\mathcal{M}}(x)/\text{Zlib}(x)$
DetectLLM (Su et al., 2023)	Detection	$\mathbb{E}_{\tilde{x}\sim\phi(x)}[\log R_{\mathcal{M}}(\tilde{x})]/(\log R_{\mathcal{M}}(x))$
ReCall (Xie et al., 2024)	MIA	$\mathbb{E}_{\tilde{x}\sim\phi(x)}[\log P_{\mathcal{M}}(\tilde{x})]/(\log P_{\mathcal{M}}(x))$
DC-PDD (Zhang et al., 2024)	MIA	$\mathbb{E}_{\tilde{x}\sim\mathcal{M}}[-\log P_{\mathcal{M}_{\text{ref}}}(\tilde{x})]$
Binoculars (Hans et al., 2024)	Detection	$(\log P_{\mathcal{M}}(x))/\mathbb{E}_{\tilde{x}\sim\mathcal{M}}[\log P_{\mathcal{M}_{\text{ref}}}(\tilde{x})]$
DetectGPT (Mitchell et al., 2023)	Detection	$\mathbb{E}_{\tilde{x}\sim\phi(x)}[\log(P_{\mathcal{M}}(x)/P_{\mathcal{M}}(\tilde{x}))]$
Neighborhood (Mattern et al., 2023)	MIA	$\mathbb{E}_{\tilde{x}\sim\phi(x)}[\log(P_{\mathcal{M}}(x)/P_{\mathcal{M}}(\tilde{x}))]$
Fast-DetectGPT (Bao et al., 2024)	Detection	$\Phi(\mathbb{E}_{\tilde{x}\sim\phi(x)}[\log(P_{\mathcal{M}}(x)/P_{\mathcal{M}}(\tilde{x}))])$
Min-K% (Shi et al., 2024)	MIA	$\frac{1}{k}\sum_{i\in\text{min-}k\%}(-\log P_{\mathcal{M}}(x_i))$
Min-K%++ (Zhang et al., 2025)	MIA	$\frac{1}{k}\sum_{i\in\text{min-}k\%}\Phi(-\log P_{\mathcal{M}}(x_i)+\mathbb{E}_{\tilde{x}_i\sim\mathcal{M}}[\log P_{\mathcal{M}}(\tilde{x}_i)])$
Lastde (Xu et al., 2025)	Detection	$(-\log P_{\mathcal{M}}(x))/\text{StdDev}(\{\text{DE}(x,\tau)\}_{\tau=1}^{\tau'})$
Lastde++ (Xu et al., 2025)	Detection	$\Phi(\text{Lastde}(x)-\mathbb{E}_{\tilde{x}\sim\phi(x)}[\text{Lastde}(\tilde{x})])$

Approximation via Text Sampling. The second approach leverages text sampling to approximate the true distribution $P_{\mathcal{Q}}$. The likelihood under the true distribution is approximated by the expected likelihood of multiple perturbations $\tilde{x}$ of the target text x,

$$P_{\mathcal{Q}}(x) \approx \mathbb{E}_{\tilde{x}\sim\phi(\cdot|x)}[P_{\mathcal{M}}(\tilde{x})].$$

where $\phi(\cdot|x)$ is a perturbation function that samples variations of the target text x. This strategy is employed by the *Neighborhood attack* (Mattern et al., 2023) from MIAs, *DetectGPT* (Mitchell et al., 2023), and *Fast-DetectGPT* (Bao et al., 2024) from machine text detection. Here, the expected likelihood of multiple perturbations can be seen as an approximate marginalization over semantically equivalent paraphrases. When the average delta of likelihood between these paraphrases is larger in $P_{\mathcal{M}}$ than it is in $P_{\mathcal{Q}}$, as assumed in *DetectGPT*, applying this marginalization to a particular datapoint would result in a better approximation of the true distribution.

Implications of Transferability. Our proof shows that the optimal test statistic is identical for both tasks, despite their different objectives. Moreover, many methods from both tasks can be interpreted as approximations of the same underlying likelihood ratio, differing in how to approximate the population distribution $P_{\mathcal{Q}}$. This fundamental connection naturally leads to methods that share similar functional forms. For example, the *Neighborhood attack* for MIAs can be viewed as a finite-sample approximation of *DetectGPT* for detection.

Importantly, our theory does not rely on any specific method pair. Instead, it provides a general explanation for why such similarities arise and why methods can transfer across tasks. Table 1 provides a breakdown of the tested methods in our work and, where applicable, includes a reformulation to fit the categorization. We discuss in more detail in Appendix B the steps we took to get to this general formulation for each example in the author's own notation.

2.4 Discussion

While our general formulation covers many methods from both tasks, there are methods that fall outside of our core framing—most notably, the methods that are not ratios but single quantities (e.g., *Min-K%*, *DC-PDD*). For these methods, we assess transferability empirically and defer a more fully unified theoretical investigation to future work. In addition, while most methods exhibit a large degree of transferability, there are some methods whose transferability is less strong. Of particular interest is the *Zlib* method which, despite being an approximate likelihood ratio, performs relatively poorly on machine text detection. We discuss some hypotheses for why this is the case in more detail in §3.3.

3 Quantifying Transferability between MIAs and MGTD

3.1 Experimental Setup

Membership Inference. We evaluate MIAs on the MIMIR dataset (Duan et al., 2024), a large-scale MIA benchmark consisting of **5 domains**[2] in the Pile (Gao et al., 2020): Wikipedia (knowledge), Pile CC (general web), PubMed Central and ArXiv (academic), HackerNews (dialogue). Members and non-members are sampled from the training and test sets of the Pile, and 13-gram filtering is used to ensure no leakage. We target the Pythia suite: **5 models** of Pythia (Biderman et al., 2023) with 160M, 1.4B, 2.8B, 6.7B, and 12B parameters.

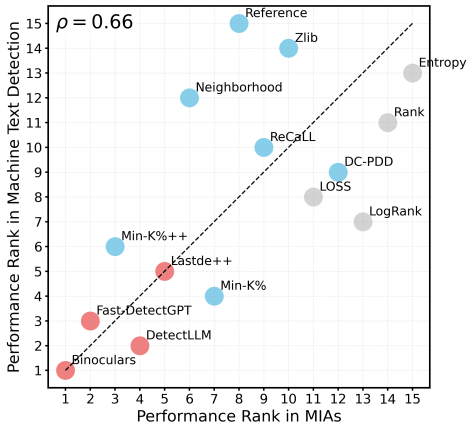

Figure 2: Relationship between method rankings across MIAs and machine text detection. Blue, red, and gray plots: MIA methods, machine text detectors, and general baselines. The dashed line denotes equal ranks. Lower rank indicates better performance on each task.

Machine Text Detection. We evaluate detection on the RAID dataset (Dugan et al., 2024), a large-scale detection benchmark consisting of generated text and human-written text in **8 domains**: Wikipedia and News (knowledge), Abstracts (academic), Recipes (instructions), Reddit (dialogue), Poetry (creative), Books (narrative), Reviews (opinions), and **5 models**: GPT-2-XL (Radford et al., 2019), MPT-30B-Chat (Team, 2023), LLaMA-2-70B-Chat (Touvron et al., 2023) as open-source models and ChatGPT (OpenAI, 2023) and GPT-4 (OpenAI et al., 2024) as closed-source models.

Evaluation Measures. To assess empirical transferability, we compute the performance ranking of all unique methods[3] on both tasks using **AUROC** and report **Spearman's rank correlation** coefficient to judge how closely the two rankings match. This choice is motivated by our proof: since both tasks share the same optimal statistic, methods would be expected to preserve their relative rankings across tasks. In contrast, absolute performance is not directly comparable across tasks since their underlying data distributions inherently differ. A high rank correlation implies that the state-of-the-art methods for one task also perform competitively on the other. Here, we use AUROC for both tasks, as it reflects a method's overall performance and provides a more reliable measure of transferability than metrics based on a single threshold.

Methods Tested. We consider **7 MIA methods**: *Reference* (Carlini et al., 2021), *Zlib* (Carlini et al., 2021), *Neighborhood attack* (Mattern et al., 2023), *Min-K% Prob* (Shi et al., 2024), *Min-K%++* (Zhang et al., 2025), *ReCaLL* (Xie et al., 2024), and *DC-PDD* (Zhang et al., 2024), **5 detectors**: *DetectGPT* (Mitchell et al., 2023), *Fast-DetectGPT* (Bao et al., 2024), *Binoculars* (Hans et al., 2024), *DetectLLM* (Su et al., 2023), and *Lastde++* (Xu et al., 2025), and **4 general baselines**: *Loss*, *Rank*, *LogRank*, and *Entropy*. An overview of the methods tested can be found in Table 1, more details on each method are in Appendix B, and the configurations are in Appendix C.

As detailed in §4, we focus on methods that do not require access to model internals, reflecting the limited access available for most modern LLMs. We also exclude supervised detectors because they rely on stylistic or semantic differences between human-written and machine-generated texts, which are different signals from model memorization that MIAs examine.

Detection Scenarios. For our main rank correlation result, we target a **white-box setting** for both MIAs and machine text detection, where token probability distributions of target models are accessible. This allows us to compare the two tasks under a common condition and is consistent with our theoretical formulation, which relies on signals derived from target model probabilities.

[2]To compare the two tasks under a common condition, we focus on textual domains as in machine text detection. See Appendix D for full results on MIAs, including technical domains: GitHub and DM Mathematics.

[3]The *Neighborhood attack* and *DetectGPT* share the same formulation and are treated as a single method.

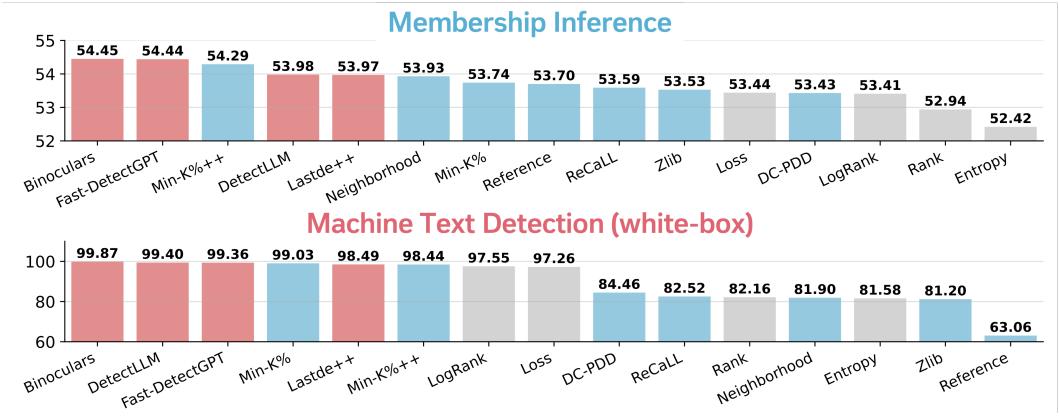

Figure 3: AUROC scores for MIA methods (blue bars) and machine text detectors (red bars) across both tasks. **Top**: MIA results on the MIMIR benchmark with 13-gram de-duplication filtering, averaged over *five domains* and *five models*. **Bottom**: Machine text detection results in a white-box setting on the RAID benchmark, averaged over *eight domains* and *three models*. Both MIA methods and machine text detectors have comparable performance in the cross-task setting. Notably, *Binoculars* achieves the best average performance on both tasks. See Appendix D for full results.

To further examine transferability in real-world scenarios, we additionally investigate the **black-box setting** for machine text detection, targeting closed-source models such as ChatGPT and GPT-4.[4] In this case, where the target model's logits are not available, we employ surrogate models (PYTHIA-160M (Biderman et al., 2023) and Llama-3-3.2B (Grattafiori et al., 2024)) and report the average detection performance across the surrogates.

3.2 MAIN RESULTS

Substantial Rank Correlation Between MIAs and Machine Text Detection. Figure 2 illustrates the relationship between the rankings of all methods when evaluated on MIAs and on machine text detection. The rankings are based on average performance on MIAs (across *five domains* and *five target models*) and machine text detection (across *eight domains* and *three generators*). We compute the rank correlation over all 15 methods and obtain a statistically significant Spearman's correlation of $\rho = 0.66$ ($p < 0.01$). This result indicates that many methods originally proposed for MIAs also perform well in machine text detection, and vice versa. Notably, when focusing on stronger methods (top-10 on MIAs), we observe an even stronger correlation of $\rho = 0.78$ ($p < 0.01$). These findings align with our theoretical framework, suggesting that methods achieving strong performance on one task, by better approximating the shared optimal metric, tend to exhibit higher transferability and perform well on the other task.

Superior Performance from the Other Task. Figure 3 shows the AUROC performance of all methods evaluated on MIAs, and their performance evaluated on machine text detection. The results on MIAs are averaged across *five domains* and *five target models*, while those on machine text detection are averaged across *eight domains* and *three generators*. Notably, we observe that *Binoculars*, originally proposed as a machine text detector, achieves the best average performance on both tasks. This suggests that current evaluations in MIAs may be biased by overlooking stronger methods from machine text detection, potentially leading to conclusions that miss valuable insights. Conversely, in machine text detection, methods from MIAs already demonstrate competitive or even superior performance. These findings call for greater cross-task awareness and development.

[4]Since the training data of such closed-source models is not accessible, true ground truth for MIAs is inherently not feasible in this setting. Therefore, MIAs are evaluated only in a white-box setting.

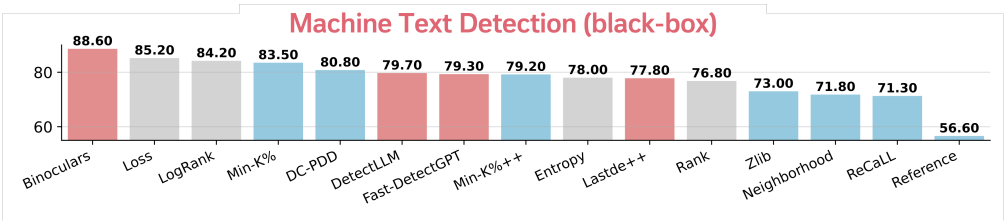

Figure 4: AUROC scores for MIA methods (blue bars) and machine text detectors (red bars) on machine text detection in a black-box setting on the RAID benchmark, averaged over *eight domains* and *two closed-source models*. Even in this real-world scenario, MIA methods such as *Min-K%* achieve performance on par with or even better than strong detectors. See Appendix D for full results.

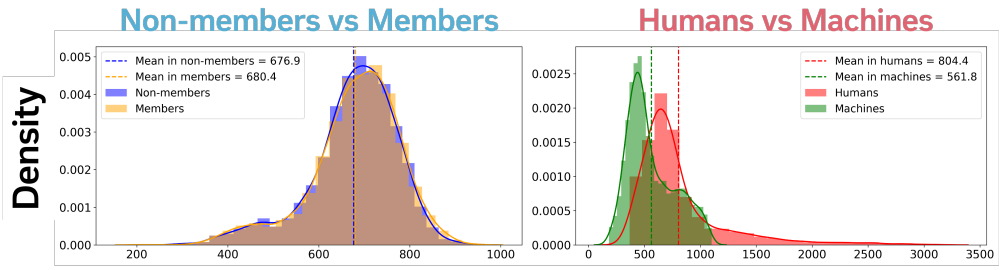

Figure 5: Zlib compression entropy distribution in MIAs and machine text detection, averaged over 3,000 randomly sampled texts for each dataset. The entropy converges between classes in MIAs but diverges in detection with a long tail, due to different prior distributions.

3.3 Analysis

Transferability in Real-World Scenarios. We further assess transferability in real-world scenarios by evaluating how well MIA methods perform in a black-box setting of machine text detection on texts generated by ChatGPT and GPT-4. Figure 4 shows the average AUROC performance of all methods from both tasks across *eight domains* and the *two generators*. While *Binoculars* still outperforms other methods by a large margin, MIA methods such as *Min-K%* and *DC-PDD* achieve performance on par with or even better than strong detectors. These results provide promising evidence of the transferability of MIAs to machine text detection in real-world scenarios.

Similar Prediction Score Distributions across Methods. To further illustrate the transferability between the two tasks, we compare the prediction score distributions of an MIA method and a detector when applied to each task. Figure 1 presents the distributions of *Min-K%++* and *Binoculars*, which demonstrate strong cross-task performance. The domain in both tasks is Wikipedia, and the target model or generator is PYTHIA-12B. We quantify the similarity in distributional shape by computing the Jensen–Shannon distance between the distributions of the two methods within each task: 0.14 for MIAs and 0.11 for detection. These small distances indicate that the two methods produce closely aligned distributions within both tasks, providing further evidence of the transferability.

***Zlib* as an Outlier Illustrates a Task Difference.** We take *Zlib* as an example of limited transferability. *Zlib* ranks 10th out of 15 methods in MIAs but drops to 14th in detection, where it calibrates the loss by dividing it by the zlib compression entropy. In MIAs, both classes are drawn from the same human-written text distribution, whereas this is not the case for detection. Machine-generated texts are known to be more compressible than human-written ones (Tulchinskii et al., 2023). Consistent with this, Figure 5 shows that the zlib entropy converges between classes in MIAs but diverges in detection. In detection, both the loss and the zlib entropy shift in the same direction, yielding similar *Zlib* scores across classes and little discriminative signal. Hence, while moderately effective for MIAs, *Zlib* transfers poorly to detection, highlighting a key task difference: **different prior distributions**. We leave a comprehensive analysis of other methods with limited transferability for future work.

4 Related Work

Proximity between MIAs and Machine Text Detection. Recent studies have suggested conceptual similarities between the two tasks. Shi et al. (2024) briefly noted that the *Neighborhood attack* and *DetectGPT* share a similar formulation based on local curvature estimation. Naseh & Mireshghallah (2025) offered intuitive explanations based on their shared reliance on likelihood signals, and empirically observed that MIAs often misclassify machine-generated non-members as members.

However, these studies remain limited to specific observations, lack a formal characterization of the underlying connection, and do not directly evaluate methods from both tasks on each standard benchmark. Our work moves beyond these conceptual similarities to establish a comprehensive theoretical and empirical foundation for transferability. We theoretically prove that both tasks share the same optimal statistic, providing a principled explanation of when and why transferability arises. Our unified evaluations empirically show substantial cross-task transfer. We notably find that a detector achieves the strongest performance among evaluated methods on both tasks, indicating that transferability is practically significant.

Membership Inference. Membership inference attacks (MIAs) aim to determine whether a given sample was part of a given model's training data (Shokri et al., 2017). In the context of language models, recent studies have applied MIAs to pretraining data detection. Prior work (Carlini et al., 2022; Watson et al., 2022) trains shadow models on random samples from a target model's pretraining data distribution. However, they are impractical for LLMs because the pretraining distribution is rarely available, and training would be too expensive. Other studies (Nasr et al., 2019; Jagannatha et al., 2021) use gradients or activations. While they are useful for exploring upper-bound MIA performance, such internal information is inaccessible for most modern LLMs without full access to the model parameters and massive computational costs. Thus, most recent MIA studies on LLMs have focused on methods that rely only on model outputs, which is the focus of our study. Since language models tend to show higher likelihood on members compared to non-members, such studies use statistical features of a target model: likelihood (Carlini et al., 2021), likelihood calibrated by another language model (Carlini et al., 2021), negative log probability curvature (Mattern et al., 2023), and average log-likelihood of the $k\%$ tokens with lowest probabilities (Shi et al., 2024). Other MIA methods are detailed in Appendix B.2.

Machine Text Detection. Many studies have investigated supervised methods, including training a classifier with human-written and machine-generated texts with labels. Supervised classifiers aim to capture stylistic or semantic differences between human-written and machine-generated texts, with simple n-gram features (Ippolito et al., 2020; Crothers et al., 2023; Verma et al., 2024) or neural representations (Li et al., 2024; Wang et al., 2024). Since MIAs fundamentally examine how much a target model memorizes a text by exploiting signals from the target model, such as likelihood, our investigation of the transferability focuses on zero-shot detectors that similarly rely on statistical features from the model. In contrast, supervised detectors capture fundamentally different signals (e.g., stylistic or semantic differences) and thus fall outside the scope of our study. Zero-shot detectors leverage statistical features, including entropy (Lavergne et al., 2008), likelihood (Solaiman et al., 2019), negative curvature of log probabilities (Mitchell et al., 2023), and cross-model entropy (Hans et al., 2024). Other zero-shot detectors are detailed in Appendix B.3.

5 Conclusion

We comprehensively study the transferability between MIAs and machine text detection. Our theoretical and empirical investigations reveal that 1) Many methods from both tasks that exhibit high transferability can be reduced to a general formulation that measures the discrepancy between the likelihood of a text under a target model and under the true distribution, reflecting their shared objective to approximate the true distribution and contributing to the transferability and 2) Many methods originally proposed for MIAs perform well in machine text detection, and vice versa, as evidenced by a strong rank correlation across the tasks and 3) Notably, a machine text detector achieves the strongest performance among evaluated methods on both tasks, demonstrating the practical impact of the transferability. These findings call for greater cross-task developments and fair evaluation across the two research communities.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting, translating, and proofreading the manuscript, improving the clarity of proofs, and writing code. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, develop proofs, or propose or refine hypotheses. We have reviewed all AI-assisted work manually and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Data Collection and Licenses. Our work uses publicly available datasets, such as MIMIR (Duan et al., 2024) for MIAs and RAID (Dugan et al., 2024) for machine text detection, which are released under clear licensing agreements. We adhere to the intended use of these dataset licenses, and note that our research involves no human subjects.

Societal Implications. Our work aims to establish theoretical and empirical foundations for transferability between MIAs and machine text detection. It highlights the critical need for cross-task awareness and unified evaluation to build stronger defenses against AI privacy risks and misuse. Conversely, such advances could be exploited to optimize evasion strategies against these privacy auditing and detection systems, stimulating further development of more robust methods.

REPRODUCIBILITY STATEMENT

We release the code, datasets, and config files used in this work at `https://anonymous.4open.science/r/mint-B44A/`, enabling others to reproduce our results. We also elaborate on the implementation details of the methods tested in Appendix C.

REFERENCES

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430, 2023.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, pp. 2633–2650, 2021.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914, 2022. doi: 10.1109/SP46214.2022.9833649.

Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. Art or artifice? Large language models and the false promise of creativity. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024. ISBN 9798400703300.

Evan Crothers, Nathalie Japkowicz, and Herna Viktor. Machine generated text: A comprehensive survey of threat models and detection methods. *arXiv preprint arXiv:2210.07321*, 2023.

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

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 12463–12492, 2024.

Futurism. CNET secretly used AI on articles that didn't disclose that fact, staff say, 2023. URL https://futurism.com/cnet-ai-articles-label. Accessed: 2025-07-10.

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. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Josh A Goldstein, Jason Chao, Shelby Grossman, Alex Stamos, and Michael Tomz. How persuasive is AI-generated propaganda? *PNAS Nexus*, 3(2):pgae034, 02 2024. ISSN 2752-6542. doi: 10.1093/pnasnexus/pgae034.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

The Guardian. Revealed: Thousands of UK university students caught cheating using AI, 2025. URL https://tinyurl.com/guardian-AI-cheating. Accessed: 2025-07-10.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting LLMs with Binoculars: Zero-shot detection of machine-generated text. In *The Forty-first International Conference on Machine Learning (ICML)*, 2024.

David A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. Membership inference attack susceptibility of clinical language models. *arXiv preprint arXiv:2104.08305*, 2021.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. Detecting fake content with relative entropy scoring. In *Proceedings of the ECAI'08 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, CEUR Workshop Proceedings, 2008.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 346–363, 2023.

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

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 24950–24962, 2023.

John X. Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G. Edward Suh, Alexander M. Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. How much do language models memorize? *arXiv preprint arXiv:2505.24832*, 2025.

Ali Naseh and Niloofar Mireshghallah. Synthetic data can mislead evaluations: Membership inference as machine text detection. *arXiv preprint arXiv:2501.11786*, 2025.

Milad Nasr, Reza Shokri, and Amir Houmansadr. 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)*, pp. 739–753, 2019. doi: 10.1109/sp.2019.00065.

Jerzy Neyman and Egon S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

OpenAI. Introducing ChatGPT, 2023. URL `https://openai.com/blog/chatgpt`. Accessed: 2023-05-10.

OpenAI, Josh Achiam, Steven Adler, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. URL `https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf`. Accessed: 2024-11-15.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

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

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics (ACL)*, pp. 12395–12412, 2023.

MosaicML NLP Team. Introducing MPT-30B: Raising the bar for open-source foundation models, 2023. URL `www.mosaicml.com/blog/mpt-30b`. Accessed: 2023-06-22.

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Eduard Tulchinskii, Kristian Kuznetsov, Kushnareva Laida, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. Intrinsic dimension estimation for robust detection of AI-generated texts. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2024.

Xianzhi Wang, Shubin Si, and Yongbo Li. Multiscale diversity entropy: A novel dynamical measure for fault diagnosis of rotating machinery. *IEEE Transactions on Industrial Informatics*, 17(8): 5419–5429, 2021. doi: 10.1109/TII.2020.3022369.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. M4GT-Bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022.

Boyi Wei, Weijia Shi, Yangsibo Huang, Noah A Smith, Chiyuan Zhang, Luke Zettlemoyer, Kai Li, and Peter Henderson. Evaluating copyright takedown methods for language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 139114–139150, 2024.

Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. ReCaLL: Membership inference via relative conditional log-likelihoods. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8671–8689, 2024.

Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. Training-free LLM-generated text detection by mining token probability sequences. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Ruth Petzold, William Yang Wang, and Wei Cheng. A survey on detection of LLMs-generated content. In *Findings of the Association for Computational Linguistics (EMNLP)*, pp. 9786–9805, 2024.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-K%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. Pretraining data detection for large language models: A divergence-based calibration method. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5263–5274, 2024.

A PROOF THAT $s = P_{\mathcal{M}}(x)$ IS SUFFICIENT

In our proof of Theorem 2.2, we transform the likelihood ratio test for the membership inference task as follows:

$$H_0 : \mathcal{M} \leftarrow \mathcal{T}(\mathcal{D}), \quad H_1 : \mathcal{M} \leftarrow \mathcal{T}(\mathcal{D} \cup \{x\}).$$

$$\Lambda_{\text{MI}}(x) = \frac{P(M = \mathcal{M}|H_1)}{P(M = \mathcal{M}|H_0)} = \frac{P(s = P_{\mathcal{M}}(x)|H_1)}{P(s = P_{\mathcal{M}}(x)|H_0)}$$

where $s = P_M(x)$ is a random variable representing the distribution of the likelihood of x under M. For this to be valid, we must show that s is sufficient for M. In other words, we must show that

$$P(M \mid s, H_0) = P(M \mid s, H_1)$$

Let $\mathcal{M}$ be a model following our asymptotic assumptions laid out in Theorem 2.2 and let $N = |\mathcal{D}|$ be the size of its training set. The likelihood of each text $y \in \mathcal{D}$ is defined as

$$P_{\mathcal{M}}(y) = \frac{1}{N}\text{count}(y, \mathcal{D})$$

where $\text{count}(y, \mathcal{D})$ is the number of times a particular text y occurs in $\mathcal{D}$. Let $\hat{\boldsymbol{p}}$ be the vector of likelihoods $P_M(y)$ for all $y \in \mathcal{D}$. This vector contains all information needed to reproduce the full output distribution of $\mathcal{M}$.[5] Thus, $\hat{\boldsymbol{p}}$ is sufficient for M and $P(M \mid \hat{\boldsymbol{p}}, H_0) = P(M \mid \hat{\boldsymbol{p}}, H_1)$.

[5]Note here that we assume models M_1 and M_2 are equivalent if $P_{M_0}(x) = P_{M_1}(x) \ \forall \ x \in \mathcal{X}$.

Now, let $\hat{\boldsymbol{p}}_{\neg x}$ be the vector of likelihoods for all elements not equal to x. In order to show that s is sufficient, we must show that $\hat{\boldsymbol{p}}_{\neg x}$ is distributed identically regardless of the hypothesis. In other words, we want to show that:

$$P_M(y \mid s, H_0) = P_M(y \mid s, H_1) \;\; \forall \;\; y \neq x, \; y \in \mathcal{D}$$

To do this, we first note the difference between our two hypotheses: For H_0, $\mathcal{D}$ was constructed by taking N uniform draws from $P_{\mathcal{Q}}$ and, for H_1, $\mathcal{D}$ was constructed by taking $N-1$ uniform draws from $P_{\mathcal{Q}}$ and one fixed draw of x. Let $N - \text{count}(x, \mathcal{D})$ be the total number of draws in $\mathcal{D}$ allocated to texts that are not x. We note that, regardless of whether we are in H_0 or H_1, these draws are distributed identically. Thus, for any $y \neq x$ we can model the frequency that y appears in $\mathcal{D}$ via the binomial

$$\text{count}(y, \mathcal{D}) \sim \text{Binomial}(N - \text{count}(x, \mathcal{D}), P_{\mathcal{Q}})$$

Substituting in likelihoods we get that:

$$P_{\mathcal{M}}(y) \sim \frac{1}{N}\text{Binomial}(N - N \cdot P_{\mathcal{M}}(x), P_{\mathcal{Q}})$$

This means that, conditioned on fixed knowledge of $s = P_{\mathcal{M}}(x)$, we get that:

$$P_{\mathcal{M}}(y \mid s, H_0) = P_{\mathcal{M}}(y \mid s, H_1) \;\; \forall \;\; y \neq x, \; y \in \mathcal{D}$$

which implies that $P(M \mid s, H_0) = P(M \mid s, H_1)$ and thus s is a sufficient statistic for our hypothesis test. □

Notes on White-Box MIA Methods. As noted in §4, we consider an MIA setting in which we do not have access to the target model's training details and parameters. This setting is practically motivated because such internal data is inaccessible for most modern LLMs. Nevertheless, in non-asymptotic and white-box settings, such internal information (e.g., activations or gradients) can capture training dynamics better than a model's likelihood alone, thereby improving membership inference and helping to explore upper-bound MIA performance, which is consistent with existing MIA literature. The white-box/black-box equivalence thus holds only in the asymptotic regime.

B Details of Methods

B.1 Baselines

(1) ***Loss*** simply uses the target sample x's loss against the model $\mathcal{M}$: $f(\mathbf{x}; \mathcal{M}) = \mathcal{L}(\mathbf{x}; \mathcal{M})$. The hypothesis is that members and machine-generated texts will have a higher likelihood on average than non-members and human-written texts.

(2) ***Entropy*** measures the expected likelihood of the next token given the preceding tokens at each time step under the model's distribution. $f(\mathbf{x}; \mathcal{M}) = \mathbb{E}_{\tilde{x} \sim \mathcal{M}}[\mathcal{L}(\tilde{x}; \mathcal{M})]$. The hypothesis is that machine-generated texts and member texts will have low entropy as the model "understands" the context more and can better model the next token distribution.

(3) ***Rank*** measures the average rank of the next token in the model's probability distribution at each time step. $f(x; \mathcal{M}) = \frac{1}{n}\sum_{i=1}^{n} \text{Rank}(x_i; \mathcal{M})$. The hypothesis is that generated text and member text will have a higher average rank than human-written text.

(4) ***LogRank*** measures the average log rank of the next token: $f(x; \mathcal{M}) = \frac{1}{n}\sum_{i=1}^{n} \log(\text{Rank}(x_i; \mathcal{M}))$. This metric smooths out contributions from very low rank tokens to reflect the probabilistic nature of rank information. Similarly to the previous metric, the hypothesis is that generated text and member text have a high average log rank.

B.2 Membership Inference Attacks

(1) ***Reference*** (Carlini et al., 2021) uses the difference in the target sample x's loss between the model $\mathcal{M}$ and another reference model $\mathcal{M}_{\text{ref}}$. We follow the original author's implementation by taking a smaller-size reference model for each of the models we tested: $f(\mathbf{x}; \mathcal{M}) = \mathcal{L}(\mathbf{x}; \mathcal{M}) - \mathcal{L}(\mathbf{x}; \mathcal{M}_{\text{ref}})$. In our notation, this can be represented as $-\log(P_{\mathcal{M}}(x)/P_{\mathcal{M}_{\text{ref}}}(x))$. This method falls into the external reference category of likelihood ratio approximation.

(2) ***Zlib*** (Carlini et al., 2021) employs the ratio of $\mathcal{L}(\mathbf{x}; \mathcal{M})$ and the zlib compression score of the target sample x: $f(\mathbf{x}; \mathcal{M}) = \mathcal{L}(\mathbf{x}; \mathcal{M}) \;/\; \text{zlib}(\mathbf{x})$. The zlib compression score is computed by constructing a dictionary of repeated substrings from the text and encoding each dictionary entry into a string of bits using Huffman Coding (Huffman, 1952). The compression rate is thus a representation of the entropy of the empirical substring distribution of the text. Thus the *Zlib* method can be equivalently rewritten as $f(\mathbf{x}; \mathcal{M}) = -\log P_{\mathcal{M}}(\mathbf{x}) \;/\; \mathbb{E}_{x \sim \mathcal{M}_{\text{ref}}}[-\log P_{\mathcal{M}_{\text{ref}}}(\mathbf{x})]$ where $\mathcal{M}_{\text{ref}}$ represents that empirical substring distribution. We see here that this method falls into the external reference category of likelihood ratio approximation.

(3) ***Neighborhood attack*** (Mattern et al., 2023) compares $\mathcal{L}(\mathbf{x}; \mathcal{M})$ to the average loss of *neighborhood samples* $\tilde{\mathbf{x}}$, which are samples crafted by perturbing the target sample x: $f(\mathbf{x}; \mathcal{M}) = \mathcal{L}(\mathbf{x}; \mathcal{M}) - \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\tilde{\mathbf{x}}^{(i)}; \mathcal{M})$. It hypothesizes that members will show a lower loss compared to their neighborhood samples. Since the quantity $\mathbb{E}_{\tilde{x} \sim \phi(x)}[\log(P_{\mathcal{M}}(x)/P_{\mathcal{M}}(\tilde{x}))]$, we consider this method to be in the text sampling category of likelihood ratio approximation.

(4) ***Min-K%*** (Shi et al., 2024) calculates the average of log-likelihood of the $k\%$ tokens with lowest probabilities: $f(\mathbf{x}; \mathcal{M}) = \frac{1}{k}\sum_{x_i \in \text{min-}k(\mathbf{x})} \log p(x_i \mid x_{<i}; \mathcal{M})$. The intuition is that members will include fewer outlier tokens with low probability compared to non-members.

(5) ***Min-K%++*** (Zhang et al., 2025) computes the average of the log-likelihood of the $k\%$ tokens with lowest probabilities, where each value is standardized over the model's vocabulary: $f(\mathbf{x}; \mathcal{M}) = \frac{1}{k}\sum_{x_i \in \text{min-k\%}} (\log p(x_i \mid x_{<i}; \mathcal{M}) - \mu_{x_{<i}})/\sigma_{x_{<i}}$. In our notation, this can be written as, $\frac{1}{k}\sum_{i \in \text{min-}k\%} \Phi(-\log P_{\mathcal{M}}(x_i) + \mathbb{E}_{\tilde{x}_i \sim \mathcal{M}}[\log P_{\mathcal{M}}(\tilde{x}_i)])$, for brevity in our Table 1, we use $\Phi(x) = x/\sigma_x$ as shorthand for the normalization by the standard deviation. Intuitively, this metric measures how unexpectedly surprising a particular token is over the vocabulary distribution.

(6) ***ReCaLL*** (Xie et al., 2024) computes the relative conditional log-likelihood between x and $P \oplus x$ where P is a set of non-member examples $P = p_1 \oplus \cdots \oplus p_n$. It hypothesizes that non-members will have lower log-likelihoods than members, given a non-member context. We represent this in our table as $\mathbb{E}_{\tilde{x} \sim \phi(x)}[\log P_{\mathcal{M}}(\tilde{x})]/(\log P_{\mathcal{M}}(x))$ where $\phi(x) = P \oplus x$. We consider this method to be in the likelihood ratio by text sampling category.

(7) ***DC-PDD*** (Zhang et al., 2024) computes the cross-entropy between the token likelihoods under the model $\mathcal{M}$ and the empirical laplace-smoothed unigram token frequency distribution under some reference corpus $\mathcal{D}'$. In the authors' notation this is $f(\mathbf{x}; \mathcal{M}) = -\frac{1}{n}\sum_{i=1}^{n} p(x_i; \mathcal{M}) \cdot \log p(x_i; \mathcal{D}')$ where $p(x_i; \mathcal{D}') = \frac{\text{count}(x_i)+1}{N'+|V|}$. In our table, we represent this metric equivalently as $\mathbb{E}_{\tilde{x} \sim \mathcal{M}}[-\log P_{\mathcal{M}_{\text{ref}}}(\tilde{x})]$ where $\mathcal{M}_{\text{ref}}$ represents the unigram token frequency distribution under $\mathcal{D}'$.

B.3 Machine Text Detectors

(1) ***DetectGPT*** (Mitchell et al., 2023) computes the degree to which the log-likelihood function under the suspected model has negative curvature for the given target text. They do this by perturbing the sequence using a T5 model Raffel et al. (2019) and evaluating the change in probability. The functional form looks like $f(x; \mathcal{M}) = \mathbb{E}_{\tilde{x} \sim \phi(x)}[\log(P_{\mathcal{M}}(x)/P_{\mathcal{M}}(\tilde{x}))]$. We directly lift this notation for use in our Table 1. This metric is an example of a text sampling based likelihood ratio approximation.

(2) ***Fast-DetectGPT*** (Bao et al., 2024) forgoes the expensive perturbation approach used by *DetectGPT* in favor of using the token likelihoods of the perturbation model directly to compute the expectation. They also divide by the mean of sample variances to further smooth out the metric. The full formulation is

$$f(x; \mathcal{M}) = \frac{\log p(x; \mathcal{M}) - \mathbb{E}_{\tilde{x} \sim q(x)}[\log p(\tilde{x}; \mathcal{M})]}{\sqrt{\mathbb{E}_{\tilde{x} \sim q(x)}[(\log p(\tilde{x}; \mathcal{M}) - \mathbb{E}_{\tilde{x} \sim q(x)}[\log p(\tilde{x}; \mathcal{M})])^2]}}$$

We report this using the shorthand $\Phi(x) = x/\sigma_x$ in our Table 1 to represent this quantity. We consider this metric another example of the text sampling based likelihood ratio approach.

(3) ***Binoculars*** (Hans et al., 2024) computes the ratio of the perplexity to the cross entropy of the text under some reference model $\mathcal{M}_{\text{ref}}$. The full formulation using the authors' notation is as follows:

$$B_{\mathcal{M}_1, \mathcal{M}_2}(s) = \frac{\sum_{i=1}^{L} \log(\mathcal{M}_1(s)_i)}{\sum_{i=1}^{L} \mathcal{M}_1(s)_i \cdot \log(\mathcal{M}_2(s)_i)}$$

We re-formulate this objective equivalently in our own notation as:

$$(\log P_{\mathcal{M}}(x))/\mathbb{E}_{\tilde{x}\sim\mathcal{M}}[\log P_{\mathcal{M}_{\text{ref}}}(\tilde{x})]$$

The intuition behind this metric is that it measures how much more likely a given text is than what we would expect according to some reference model. We consider this metric a case of approximation via reference model.

(4) ***DetectLLM*** (Su et al., 2023) is a variant of *DetectGPT* that uses the log rank as the core quantity to test rather than the log likelihood. The authors propose two metrics, Log-Likelihood Log-Rank Ratio (LRR) and Normalized Log-Rank Perturbation (NPR). For the purposes of our work we consider the NPR metric which has superior performance and is characterized by the following formulation:

$$\text{NPR} = \frac{\frac{1}{n}\sum_{p=1}^{n}\log r_\theta(\tilde{x}_p)}{\log r_\theta(x)}$$

where $r_\theta(x_i)$ represents the rank of the token x in the model's output distribution. We reformulate this equivalently as

$$\frac{\mathbb{E}_{\tilde{x}\sim\phi(x)}[\mathcal{R}(\tilde{x};\mathcal{M})]}{\mathcal{R}(x;\mathcal{M})}$$

using the notation $\mathcal{R}(x;\mathcal{M})$ to denote the log rank similar to how $\mathcal{L}(x;\mathcal{M})$ denotes the negative log likelihood. While this metric doesn't quite fall cleanly into our likelihood ratio categorization (since it does not compute likelihood) we nonetheless note the strong similarities between this metric and other approximate likelihood ratios.

(5) ***Lastde++*** (Xu et al., 2025) utilizes a quantity known as multi-scale diversity entropy (MDE) (Wang et al., 2021) to measure the local fluctuations in likelihood across a particular text sequence. Their metric is:

$$\text{Lastde}(x;\mathcal{M}) = \frac{\mathcal{L}(x;\mathcal{M})}{\text{StdDev}(\{\text{DE}(s,\varepsilon,1),...,\text{DE}(s,\varepsilon,\tau')\})}$$

$$\text{DE}(s,\varepsilon,\tau) = -\frac{1}{\ln\varepsilon}\sum_{i=1}^{\varepsilon} P_i^{(\tau)}\ln P_i^{(\tau)}$$

Where $P_i^{(\tau)}$ measures the *diversity* of text, i.e. the extent to which adjacent segments of tokens have similar probability sequences (see Xu et al. (2025)). Intuitively the *Lastde* metric can be thought of as comparing likelihood to the expected diversity. *Lastde++* takes this quantity and applies a sampling based perturbation and normalization similar to *Fast-DetectGPT* (Bao et al., 2024).

$$\text{Lastde++}(x) = \frac{\text{Lastde}(x;\mathcal{M}) - \mathbb{E}_{\tilde{x}\sim\phi(x)}[\text{Lastde}(\tilde{x},\mathcal{M})]}{\sqrt{\mathbb{E}_{\tilde{x}\sim\phi(x)}[(\text{Lastde}(\tilde{x},\mathcal{M}) - \mathbb{E}_{\tilde{x}\sim\phi(x)}[\text{Lastde}(\tilde{x},\mathcal{M})])^2]}}$$

This can be thought of as measuring whether or not the quantity tracked by $\text{Lastde}(x)$ is at a local maximum for the particular sequence of text.

We consider neither *Lastde++* or *Lastde* to be approximating likelihood ratios as the diversity-entropy (DE) metric seems to be more intuitively thought of as measuring variance rather than likelihood. We leave to future work a more thorough analysis of these variance-based metrics.

C Implementation Details

C.1 Mint

We release a unified evaluation suite for membership inference attacks (MIAs) and machine-generated text detection: `https://anonymous.4open.science/r/mint-B44A/`. It provides a common framework for evaluation across the two tasks. Some key features of Mint include:

- Support for 15 recent methods and commonly used benchmarks across both tasks.
- A modular class design with helper functions for easy integration of new methods.
- An efficient inference pipeline with shared likelihood caching for multiple methods.

Our experiments were executed using Python 3.9.1 and PyTorch 2.5.1 (CUDA 12.6, NVIDIA Driver 560.28.03) with an Intel Xeon Silver 4314 CPU, 300GB of RAM, and four NVIDIA RTX A6000 GPUs running openSUSE Leap 15.6. The total compute cost was approximately 350 GPU-hours.

C.2 MEMBERSHIP INFERENCE ATTACKS

Reference (Carlini et al., 2021) Following the original author's implementation, we take a smaller-sized reference model for each target model. In MIAs, we use PYTHIA-70M for all target models. In white-box machine text detection, we use GPT2-Small as the smaller model for GPT2-XL, MPT-7B-Chat for MPT-30B-Chat, and LLaMA-7B-Chat for LLaMA-70B-Chat. In black-box machine text detection, we use LLaMA-3-1B as the smaller model for LLaMA-3-3.2B and PYTHIA-70M for PYTHIA-160M.

Neighborhood attack (Mattern et al., 2023) We use the repository[6] default settings, namely T5-Large as mask filling model and 0.3 as the masking rate, across MIAs, white-box machine text detection, and black-box machine text detection.

Min-K% (Shi et al., 2024) We use the setting that was found to ensure the favorable performance in the original paper, namely $k = 20$ as the percent of tokens with the lowest probabilities. Likewise, the token percentage k for *Min-K%++* (Zhang et al., 2025) is also set as $k = 20$ for fair comparison.

ReCaLL (Xie et al., 2024) We set the number of prefixes to $n = 10$, which was found to yield favorable results in the original paper. In MIAs, for each domain, we retrieve non-members as prefixes from the Pile dataset, excluding those in the MIMIR benchmark. To adapt *ReCaLL* to detection, we use human-written texts as prefixes, since they belong to the negative class. For each domain, these human-written prefixes are retrieved from a subset of the RAID that was not used in our test set.

DC-PDD (Zhang et al., 2024) Following the official GitHub repository[7], we take a subset of C4 (Raffel et al., 2019) and build a token frequency distribution with the tokenizer of each target model in MIAs and white-box machine text detection, and of each surrogate model in black-box detection.

C.3 MACHINE TEXT DETECTORS

DetectGPT (Mitchell et al., 2023) We follow the repository[8] default settings, namely T5-Large as mask filling model and 0.3 as the masking rate, across MIAs, white-box machine text detection, and black-box machine text detection.

Fast-DetectGPT (Bao et al., 2024) Following the original paper, we employ the target model as both the scoring and perturbation model in MIAs and white-box machine text detection. In black-box detection, we instead use surrogate models as those (see §3.1).

Binoculars (Hans et al., 2024) is reported to work best when the target and reference models are similar in performance. Following the original setting, we use the official code[9] to compute perplexity under a reference model. For MIAs, since the PYTHIA does not provide corresponding chat versions for each model size, we use the PYTHIA-deduped models as references. In white-box detection, we use GPT2-XL-Chat [10] for GPT2-XL, MPT-30B for MPT-30B-Chat, and LLaMA-2-70B for LLaMA-2-70B-Chat. In black-box detection, we adopt LLaMA-3.2-3B-instruct for LLaMA-3-3.2B and PYTHIA-160M-deduped for PYTHIA-160M.

DetectLLM (Su et al., 2023) In line with *Fast-DetectGPT*, we utilize the target model as both the scoring and perturbation model in MIAs and white-box detection. For black-box detection, we instead use surrogate models (see §3.1).

Lastde++ (Xu et al., 2025) In our implementation, we use the official code and use the repository[11] default settings of the sliding window size $s = 4$, the interval precision $\epsilon = 8$, and the number of scales $\tau' = 15$.

[6] https://github.com/mireshghallah/neighborhood-curvature-mia
[7] https://github.com/zhang-wei-chao/DC-PDD
[8] https://github.com/eric-mitchell/detect-gpt
[9] https://github.com/ahans30/Binoculars
[10] lgaalves/gpt2-xl_lima
[11] https://github.com/TrustMedia-zju/Lastde_Detector

D Full Performance on Membership Inference and Machine Text Detection

Table 2, Table 3, and Table 4 report the full results on membership inference, white-box machine text detection, and black-box machine text detection, respectively.

Table 2: Full comparison of membership inference performances (AUROC) of MIA methods and machine text detectors on the MIMIR benchmark with 13-gram deduplication. Target models are PYTHIA with 160M, 1.4B, 2.8B, 6.9B, and 12B parameters. Gray, blue, and red areas indicate general baseline methods, MIA methods, and machine text detectors, respectively. Textual domains are the bolded columns (Wikipedia, Pile CC, PubMed, ArXiv, HackerNews).

Method	**Wikipedia**					GitHub					**Pile CC**					**PubMed Central**				
	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	51.2	53.4	54.1	55.6	56.5	76.3	80.2	81.4	82.7	83.6	50.1	51.0	51.2	52.1	52.7	50.9	52.1	52.7	53.4	54.0
Rank	49.7	52.9	53.9	56.2	57.5	70.0	74.6	75.7	77.4	77.8	50.9	51.6	51.8	52.2	52.1	51.9	52.3	52.4	53.0	53.6
LogRank	51.3	53.5	54.3	55.6	56.9	75.9	80.0	81.2	82.5	83.3	50.3	51.0	51.2	52.3	52.7	50.8	51.8	52.4	53.3	53.7
Entropy	50.9	52.1	52.6	53.1	53.4	76.3	79.7	80.7	81.9	82.7	49.6	50.2	50.5	50.9	51.3	51.7	51.8	52.1	52.2	52.3
Reference	51.7	54.4	55.2	57.4	58.5	37.3	41.0	41.8	43.2	43.6	50.9	52.7	52.8	53.9	54.5	49.4	52.2	52.6	53.5	54.1
Zlib	50.4	53.1	54.0	55.7	56.7	79.7	82.9	83.9	85.0	85.7	51.1	52.1	52.3	53.2	53.6	51.5	52.6	53.1	53.7	54.2
Neighborhood	51.2	54.4	54.8	56.0	57.7	75.4	74.7	74.1	74.8	74.9	51.4	52.7	53.3	54.8	54.7	52.6	55.0	55.8	56.5	57.0
Min-K%	50.6	53.5	54.7	56.7	57.8	75.2	79.8	81.0	82.5	83.4	50.7	51.4	51.6	52.5	52.9	51.4	52.6	53.0	53.9	54.8
Min-K%++	51.2	55.2	56.4	59.6	60.7	73.2	78.2	79.7	81.1	82.4	50.9	52.4	52.2	54.0	54.7	50.6	52.2	52.8	54.3	55.2
ReCaLL	50.5	54.2	54.6	57.2	57.7	72.2	77.3	79.6	80.6	81.9	48.2	49.4	50.7	51.5	51.2	52.1	52.6	55.1	54.9	56.0
DC-PDD	52.4	53.9	54.5	55.8	56.4	82.1	85.2	86.2	86.9	87.6	50.8	52.6	52.7	53.3	53.6	50.5	51.7	52.5	52.9	53.2
DetectGPT	51.2	54.4	54.8	56.0	57.7	75.4	74.7	74.1	74.8	74.9	51.4	52.7	53.3	54.8	54.7	52.6	55.0	55.8	56.5	57.0
Fast-DetectGPT	51.9	54.9	56.3	60.0	62.9	57.8	67.2	69.6	71.4	72.3	51.8	54.2	53.9	55.6	56.1	49.1	51.7	52.8	55.1	56.4
Binoculars	51.7	55.2	56.7	58.5	60.6	71.8	77.2	74.5	80.3	81.7	51.2	53.6	54.5	55.1	55.0	50.4	52.4	53.0	55.2	55.9
DetectLLM	51.6	54.0	55.4	58.3	61.4	56.4	66.8	69.6	71.3	71.8	52.2	53.7	53.5	55.5	55.8	49.0	51.3	52.5	54.8	55.7
Lastde++	50.8	54.2	55.7	59.3	61.4	52.8	64.7	66.8	68.6	69.7	50.9	52.3	52.7	54.6	54.5	50.7	52.2	53.0	54.6	56.4

Method	**ArXiv**					DM Mathematics					**HackerNews**					**Avg. (textual domains)**				
	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	54.6	55.8	56.4	57.5	58.1	67.8	67.5	67.2	67.3	67.3	50.5	51.8	52.6	53.4	54.2	51.5	52.8	53.4	54.4	55.1
Rank	52.0	52.4	52.6	54.6	55.0	60.6	60.4	60.7	60.6	60.6	51.7	52.7	52.5	53.6	54.5	51.2	52.4	52.6	53.9	54.5
LogRank	54.4	55.6	56.1	57.5	57.9	66.3	66.4	66.2	66.4	66.2	50.4	51.7	52.4	53.7	54.3	51.4	52.7	53.3	54.5	55.1
Entropy	54.8	55.4	55.1	55.7	55.7	68.4	67.6	67.4	67.2	67.1	50.5	52.2	52.0	52.1	52.1	51.5	52.3	52.5	52.8	53.0
Reference	51.2	53.3	54.1	55.8	56.8	45.0	44.8	44.4	44.4	44.3	50.0	52.1	53.8	55.2	56.6	50.6	52.9	53.7	55.2	56.1
Zlib	54.2	55.3	55.7	56.7	57.2	64.6	64.7	64.6	64.6	64.6	51.2	51.9	52.4	52.9	53.4	51.7	53.0	53.5	54.4	55.0
Neighborhood	53.1	54.7	54.2	54.9	55.0	53.3	51.8	53.1	50.8	53.3	51.1	51.0	51.8	51.9	52.6	51.0	52.2	52.6	53.4	53.9
Min-K%	53.3	55.0	55.8	57.3	58.4	64.7	65.2	64.9	65.1	65.1	50.9	51.8	53.1	54.3	55.3	51.4	52.9	53.6	54.9	55.8
Min-K%++	51.3	53.8	55.9	56.9	59.9	58.8	57.9	58.7	58.4	58.2	51.1	51.5	53.1	54.6	56.4	51.0	53.0	54.1	55.9	57.4
ReCaLL	53.4	54.5	55.4	57.3	58.3	58.0	56.4	56.8	53.7	53.1	52.6	52.4	52.4	53.7	54.0	51.4	52.6	53.6	54.9	55.4
DC-PDD	54.7	56.3	56.3	57.4	57.7	63.9	63.8	63.5	63.4	63.6	49.4	51.0	51.4	52.1	52.7	51.6	53.1	53.5	54.3	54.7
DetectGPT	53.1	54.7	54.2	54.9	55.0	53.3	51.8	53.1	50.8	53.3	51.1	51.0	51.8	51.9	52.6	51.0	52.2	52.6	53.4	53.9
Fast-DetectGPT	51.5	53.2	54.9	57.4	59.3	52.4	53.1	54.2	54.0	54.7	50.1	49.5	51.9	54.1	56.4	50.9	52.7	54.0	56.4	58.2
Binoculars	54.1	54.6	54.9	57.6	59.9	55.7	53.8	53.5	52.1	52.8	49.6	50.4	51.5	53.8	55.6	51.3	53.3	54.2	56.1	57.5
DetectLLM	51.5	53.1	54.7	57.6	58.6	51.9	51.8	53.2	53.6	53.0	49.8	49.2	51.1	53.6	55.5	50.8	52.3	53.4	56.0	57.4
Lastde++	50.8	52.9	54.7	56.1	58.1	52.3	51.6	52.1	52.2	52.3	50.0	51.1	52.0	54.0	56.0	50.6	52.5	53.6	55.7	57.3

Table 3: Full comparison of white-box machine text detection performances (AUROC) of MIA methods and machine text detectors on the RAID benchmark. Target models are GPT-2: GPT-2-XL, MPT: MPT-30B-Chat, LLaMA: LLaMA-2-70B-Chat. Gray, blue, and red areas indicate general baseline methods, MIA methods, and machine text detectors, respectively.

Method	Abstracts			Books			News			Poetry		
	GPT-2	MPT	LLaMA	GPT-2	MPT	LLaMA	GPT-2	MPT	LLaMA	GPT-2	MPT	LLaMA
Loss	97.4	98.6	100.0	98.6	99.9	100.0	94.6	99.9	100.0	96.8	98.3	96.2
Rank	98.2	53.9	93.8	99.4	91.9	96.5	95.5	70.3	86.1	98.0	85.6	88.8
LogRank	98.5	96.4	100.0	99.4	99.9	100.0	96.5	99.7	100.0	97.6	98.3	95.6
Entropy	61.6	55.0	99.9	60.7	97.4	99.9	38.9	92.3	99.9	89.3	90.3	95.4
Reference	21.7	52.4	53.5	59.7	81.7	65.2	45.9	97.8	81.4	40.8	88.7	73.6
Zlib	87.1	47.0	99.9	64.1	66.8	96.0	55.8	80.4	99.7	73.4	73.9	87.6
Neighborhood	95.3	58.4	94.7	92.1	64.8	86.3	85.3	67.7	88.9	83.0	73.2	76.2
Min-K%	99.8	98.8	100.0	99.9	99.9	100.0	99.4	99.9	100.0	99.3	99.0	96.8
Min-K%++	99.6	99.6	97.1	99.8	99.0	98.2	99.7	99.5	98.9	99.4	99.1	90.3
ReCaLL	62.6	87.1	99.6	81.1	95.3	97.8	68.8	89.9	93.3	76.9	87.7	97.6
DC-PDD	48.6	95.3	100.0	64.1	99.8	99.8	47.4	99.3	99.9	82.5	97.0	94.2
DetectGPT	95.3	58.4	94.7	92.1	64.8	86.3	85.3	67.7	88.9	83.0	73.2	76.2
Fast-DetectGPT	99.7	100.0	99.8	99.8	100.0	99.9	99.7	100.0	100.0	99.3	100.0	94.4
Binoculars	99.8	99.9	100.0	99.8	100.0	100.0	99.8	99.9	100.0	99.5	100.0	100.0
DetectLLM	99.7	100.0	100.0	99.8	100.0	99.8	99.7	100.0	100.0	99.4	99.9	93.2
Lastde++	99.7	100.0	97.8	99.8	100.0	98.8	99.6	100.0	100.0	99.4	99.9	88.8

Method	Recipes			Reddit			Reviews			Wikipedia		
	GPT-2	MPT	LLaMA	GPT-2	MPT	LLaMA	GPT-2	MPT	LLaMA	GPT-2	MPT	LLaMA
Loss	70.4	100.0	100.0	97.7	97.4	99.7	97.9	99.9	99.9	91.4	100.0	99.9
Rank	82.3	48.8	17.5	98.3	73.9	64.6	98.8	82.8	95.3	97.0	73.4	81.3
LogRank	71.5	99.9	100.0	98.3	96.2	99.7	98.8	99.9	99.9	95.5	99.9	99.8
Entropy	40.2	98.4	100.0	72.3	78.3	99.7	59.0	98.6	99.9	37.6	94.5	99.1
Reference	34.9	84.7	12.6	51.8	53.2	85.4	60.4	78.0	85.9	46.8	92.6	64.9
Zlib	61.0	98.0	99.9	96.0	50.9	99.6	60.3	65.9	99.2	87.4	99.3	99.8
Neighborhood	82.5	58.2	89.2	97.5	57.0	92.1	90.6	70.8	97.2	93.7	82.4	88.6
Min-K%	89.4	100.0	100.0	99.3	96.6	99.7	99.7	99.9	100.0	99.4	100.0	99.9
Min-K%++	98.9	98.8	99.2	99.3	96.9	91.7	99.4	99.7	99.5	99.6	99.6	99.7
ReCaLL	60.2	97.1	100.0	40.5	87.8	98.6	67.4	89.4	95.8	37.0	71.1	97.9
DC-PDD	41.3	99.9	100.0	62.9	97.7	99.9	62.2	99.8	99.9	38.3	98.3	98.9
DetectGPT	82.5	58.2	89.2	97.5	57.0	92.1	90.6	70.8	97.2	93.7	82.4	88.6
Fast-DetectGPT	99.1	100.0	100.0	99.8	99.9	94.5	99.6	99.8	99.6	99.8	100.0	100.0
Binoculars	99.6	100.0	100.0	99.7	99.1	100.0	99.8	99.9	100.0	99.8	100.0	100.0
DetectLLM	99.0	100.0	99.9	99.8	99.9	96.7	99.7	99.8	99.7	99.8	100.0	99.9
Lastde++	99.0	100.0	99.4	99.8	99.6	85.1	99.5	99.6	98.0	99.9	100.0	99.9

Table 4: Full comparison of black-box machine text detection performances (AUROC) of MIA methods and machine text detectors on the RAID benchmark. Target models are ChatGPT and GPT-4. Surrogate models are Llama: LLaMA-3-3.2B, Pythia: PYTHIA-160M. Gray, blue, and red areas indicate general baseline methods, MIA methods, and machine text detectors, respectively.

Method	Abstracts				Books				News				Poetry			
	ChatGPT		GPT-4		ChatGPT		GPT-4		ChatGPT		GPT-4		ChatGPT		GPT-4	
	Llama	Pythia	Llama	Pythia	Llama	Pythia	Llama	Pythia	Llama	Pythia	Llama	Pythia	Llama	Pythia	Llama	Pythia
Loss	96.0	95.6	38.5	51.9	99.8	98.2	79.1	72.0	99.5	99.1	66.1	84.0	88.9	79.4	46.0	51.0
Rank	73.6	82.9	49.1	57.3	99.5	91.8	96.0	59.3	93.2	92.3	62.6	69.2	85.6	73.9	48.0	36.5
LogRank	96.6	96.2	40.3	53.8	99.8	97.5	77.6	65.1	99.7	98.8	68.1	80.4	88.9	76.1	44.1	43.0
Entropy	74.5	68.3	19.8	39.2	99.4	86.2	83.7	62.3	99.1	87.4	68.0	69.3	87.4	62.3	44.7	37.3
Reference	37.6	50.3	29.0	38.9	58.4	89.2	50.6	70.4	50.0	54.8	33.8	42.1	79.5	77.2	71.4	50.1
Zlib	53.3	39.7	12.4	14.8	72.5	59.8	64.1	62.2	89.6	76.9	70.5	72.7	68.2	56.0	60.2	63.4
Neighborhood	61.0	77.1	29.4	45.9	78.9	87.4	57.4	58.8	81.6	88.3	53.9	61.4	74.5	78.0	66.7	61.8
Min-K%	98.3	97.5	56.1	62.5	99.8	96.4	74.4	58.6	99.6	98.5	63.5	76.3	89.8	77.0	42.3	37.2
Min-K%++	99.1	99.3	71.9	68.7	90.3	98.9	36.7	70.4	73.6	99.7	38.0	86.3	65.5	92.4	44.4	69.0
ReCaLL	99.3	98.0	81.1	97.4	96.3	87.6	65.6	83.2	87.4	66.5	47.0	55.7	89.0	73.8	52.3	54.5
DC-PDD	88.7	85.9	38.5	62.5	99.0	96.1	77.2	86.0	93.6	85.1	47.6	72.0	87.2	80.9	51.0	64.9
DetectGPT	61.0	77.1	29.4	45.9	78.9	87.4	57.4	58.8	81.6	88.3	53.9	61.4	74.5	78.0	66.7	61.8
Fast-DetectGPT	99.3	98.5	77.0	58.7	83.9	96.1	38.0	69.9	72.2	99.4	46.5	88.0	70.9	89.7	53.2	79.0
Binoculars	100.0	98.1	96.5	94.5	99.4	97.1	77.0	70.5	98.8	78.6	83.0	51.2	98.9	85.9	57.4	73.0
DetectLLM	98.9	98.0	77.1	57.7	85.6	94.9	42.7	66.7	77.8	99.1	51.0	85.2	75.7	88.6	56.0	74.8
Lastde++	98.6	96.9	75.8	58.3	83.6	91.8	40.2	60.6	69.1	98.6	45.2	84.2	71.9	84.7	53.0	67.0

Method	Recipes				Reddit				Reviews				Wikipedia			
	ChatGPT		GPT-4		ChatGPT		GPT-4		ChatGPT		GPT-4		ChatGPT		GPT-4	
	Llama	Pythia	Llama	Pythia	Llama	Pythia	Llama	Pythia	Llama	Pythia	Llama	Pythia	Llama	Pythia	Llama	Pythia
Loss	99.7	99.0	93.4	94.8	98.8	96.6	91.6	87.0	99.9	99.5	89.0	80.9	99.0	97.1	73.8	82.4
Rank	56.9	89.7	48.0	90.5	87.7	86.2	79.0	75.7	99.1	94.7	75.7	63.1	94.8	93.3	77.2	75.4
LogRank	99.5	98.9	92.9	94.6	98.5	95.6	90.4	84.3	99.9	99.2	86.2	72.2	99.2	97.4	76.0	82.3
Entropy	99.7	92.6	93.5	83.1	96.5	88.5	90.8	82.1	99.9	93.4	91.2	66.0	98.7	83.4	73.6	72.9
Reference	55.6	36.5	36.5	29.4	60.5	83.3	52.4	74.6	76.8	94.8	58.8	82.4	52.4	50.0	41.3	42.9
Zlib	96.9	93.4	90.9	90.8	78.0	64.9	94.0	92.8	91.2	75.8	68.5	65.3	99.9	99.9	98.0	98.6
Neighborhood	66.3	87.5	58.6	78.8	76.4	86.6	74.6	72.1	94.3	87.6	72.7	55.3	92.9	91.7	71.7	68.0
Min-K%	99.5	98.7	92.8	93.9	98.3	93.9	87.5	79.6	99.9	98.1	79.5	65.4	99.5	97.9	78.9	81.2
Min-K%++	95.1	99.7	76.6	94.5	89.0	96.8	64.8	78.6	91.5	99.2	46.0	81.6	84.5	99.3	52.3	80.8
ReCaLL	95.9	88.0	69.2	74.2	97.0	77.3	58.2	43.2	87.9	65.0	58.4	67.7	80.3	28.0	34.6	22.9
DC-PDD	98.2	94.4	88.1	84.7	98.6	96.0	87.3	86.2	99.8	97.0	86.7	90.8	88.3	71.9	45.2	55.2
DetectGPT	66.3	87.5	58.6	78.8	76.4	86.6	74.6	72.1	94.3	87.6	72.7	55.3	92.9	91.7	71.7	68.0
Fast-DetectGPT	84.1	99.8	67.2	98.8	85.8	94.1	66.3	85.0	81.2	99.8	40.4	86.6	79.8	99.9	54.9	93.0
Binoculars	99.9	98.2	97.8	94.7	99.6	91.3	94.6	76.1	99.5	96.5	84.0	83.8	98.8	97.7	85.1	77.8
DetectLLM	82.2	99.8	66.6	99.1	85.9	92.6	67.4	83.6	84.6	99.7	42.2	84.8	82.1	99.9	56.8	92.6
Lastde++	83.9	99.4	67.8	97.9	83.9	90.4	66.6	83.9	84.2	99.3	40.9	80.4	81.6	99.8	57.4	92.0