Track 1:
Can Watermarking Large Language Models Prevent Copyrighted Text Generation and Hide Training Data?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large Language Models (LLMs) have demonstrated impressive capabilities in generating diverse and contextually rich text. However, concerns regarding copyright infringement arise as LLMs may inadvertently produce copyrighted material. In this paper, we first investigate the effectiveness of watermarking LLMs as a deterrent against the generation of copyrighted texts. Through theoretical analysis and empirical evaluation, we demonstrate that incorporating watermarks into LLMs significantly reduces the likelihood of generating copyrighted content, thereby addressing a critical concern in the deployment of LLMs. However, we also find that watermarking can have unintended consequences on Membership Inference Attacks (MIAs), which aim to discern whether a sample was part of the pretraining dataset and may be used to detect copyright violations. Surprisingly, we find that watermarking adversely affects the success rate of MIAs, complicating the task of detecting copyrighted text in the pretraining dataset. These results reveal the complex interplay between different regulatory measures, which may impact each other in unforeseen ways. Finally, we propose an adaptive technique to improve the success rate of a recent MIA under watermarking. Our findings underscore the importance of developing adaptive methods to study critical problems in LLMs with potential legal implications.

1 Introduction

In recent years, Large Language Models (LLMs) have pushed the frontiers of natural language processing by facilitating sophisticated tasks like text generation, translation, and summarization. With their impressive performance, LLMs are increasingly integrated into various applications, including virtual assistants, chatbots, content generation, and education. However, the widespread usage of LLMs brings forth serious concerns regarding potential copyright infringements. Addressing these challenges is critical for the ethical and legal deployment of LLMs.

Copyright infringement involves unauthorized usage of copyrighted content, which violates the intellectual property rights of copyright owners, potentially undermining content creators' ability to fund their work, and affecting the diversity of creative outputs in society. Additionally, violators can face legal consequences, including lawsuits and financial penalties. For LLMs, copyright infringement can occur through (1) generation of copyrighted content during deployment and (2) illegal usage of copyrighted works during training. Ensuring the absence of copyrighted content in the vast training datasets of LLMs is challenging. Moreover, legal debates around generative AI copyright infringement vary by region, complicating compliance further.

Submitted to AdvML-Frontiers'24: The 3nd Workshop on New Frontiers in Adversarial Machine Learning@NeurIPS'24, Vancouver, CA. Do not distribute.

Figure 1: Illustration of the effect of LLM watermarking on generation of copyrighted content. We observe that watermarking can make it more than 10^{20} times less likely for Llama-30B to generate copyrighted content.

Current lawsuits against AI companies for unauthorized use of copyrighted content (e.g., Andersen v. Stability AI Ltd, NYT v. OpenAI) highlight the urgent need for methods to address these challenges. In this paper, we focus on studying the effects of watermarking LLMs on two critical issues: (1) preventing the generation of copyrighted content, and (2) detecting copyrighted content in training data. We show that watermarking can significantly impact both the generation of copyrighted text and the detection of copyrighted content in training data.

Firstly, we observe that current LLM output watermarking techniques can significantly reduce the probability of LLMs generating copyrighted content, by tens of orders of magnitude. Our empirical results focus on two recent watermarking methods: UMD [20] and Unigram-Watermark [43]. Both methods split the vocabulary into two sets (green and red) and bias the model towards selecting tokens from the green set by altering the logits distribution, thereby embedding a detectable signal. We provide both empirical and theoretical results to support our findings.

Secondly, we demonstrate that watermarking techniques can decrease the success rate of Membership Inference Attacks (MIAs), which aim to detect whether a piece of copyrighted text was part of the training dataset. Since MIAs exploit the model's output, their performance can suffer under watermarking due to changes in the probability distribution of output tokens. Our comprehensive empirical study, including 5 recent MIAs and 5 LLMs, shows that the AUC of detection methods can be reduced by up to 16% in the presence of watermarks.

Finally, we propose an adaptive method designed to enhance the success rate of a recent MIA [31] in detecting copyright violations under watermarking. This method applies a correction to the model's output to account for the perturbations introduced by watermarks. By incorporating knowledge about the watermarking scheme, we improve the detection performance for pretraining data, counteracting the obfuscation caused by watermarking. Our contribution underscores the importance of continuously developing adaptive attack methodologies to keep pace with advances in defense mechanisms.

The rest of the paper is organized as follows. We formally introduce the problems that we study in Section 2 and present the first two contributions of empirical results in Sections 3 and 4. We discuss prior work on LLM watermarking, copyright, memorization, and membership inference in the appendix. Additional experiments, the theoretical results, the adaptive version of the Min-K% Prob membership inference attack, and a discussion on the limitations of our work are included in the appendix, too. In Section 5, we provide concluding remarks.

2 Setup and Notations

2.1 Definitions

Let D be a training dataset, C be all the copyrighted texts, and C_D be all the copyrighted texts that are part of D. We give definitions for the following setups.

Verbatim Memorization of Copyrighted Content. For a fixed $k \in \mathbb{N}$, Carlini et al. [8] defines a string s as being memorized by a model if s is extractable with a prompt p of length k using greedy decoding and the concatenation $p \oplus s \in D$. We adopt a similar definition for verbatim memorization of copyrighted content but employ a continuous metric to measure it. Specifically, we measure verbatim memorization of a text $c \in C$ using the perplexity of the model on the copyrighted text c_p when given the prefix p as a prompt (where c_p represents the text c after removing its prefix p). Note that for $c_p = c_p^{(1)} \oplus c_p^{(2)} \oplus \cdots \oplus c_p^{(n)}$ we compute the perplexity using the following formula $\texttt{perplexity}(c_p|p) = \left(\prod_{i=1}^{n} \mathbb{P}(c_p^{(i)}|p \oplus c_p^{(0)} \oplus c_p^{(1)} \oplus \cdots \oplus c_p^{(i-1)}\right)^{-\frac{1}{n}}$, where $c_p^{(0)}$ is the empty string. In our experiments, p is either an empty string or the first 10, 20, or 100 tokens of c. Lower

perplexity thereby indicate higher levels of memorization.

MIAs for Copyrighted Training Data Detection. MIAs are privacy attacks aiming to detect whether a sample was part of the training set. We define an MIA for copyrighted data as a binary classifier $A(\cdot)$, which ideally outputs $A(x) = 1, \forall x \in C_D$ and $A(x) = 0, \forall x \in C - C_D$. In practice, $A(\cdot)$ is defined by thresholding a metric (e.g., perplexity), i.e., $A(x) = 1, \forall x$ such that $\texttt{perplexity}(x) < t$ and 0, otherwise. Since the threshold t needs to be set, prior work [31] uses `AUC` (Area Under the ROC Curve) as an evaluation metric which is independent of t. Note that we employ the same metric in our experiments.

LLM Watermarking. Watermarking LLMs consists of introducing signals during its training or inference that are difficult to detect by humans without the knowledge of a *watermark key* but can be detected using an algorithm if the key is known. We focus our paper on recent methods that employ logits distribution changes as a way of inserting watermark signals during the decoding process [20, 43].

2.2 MIAs

Current MIAs for detecting training data rely on thresholding various heuristics that capture differences in output probabilities for each token between data included in the training set and data that was not. Below, we present an overview of these heuristics.

Perplexity. This metric distinguishes between data used to train the model (members) and data that was not (non-members), as members are generally expected to have lower perplexity.

Smaller Ref, Lowercase and Zlib [6]. Smaller Ref is defined as the ratio of the log-perplexity of the target LLM on a sample to the log-perplexity of a smaller reference LLM on the same sample. Lowercase represents the ratio of the log-perplexity of the target LLM on the original sample to the log-perplexity of the LLM on the lowercase version of the sample. Zlib is defined as the ratio of the log-perplexity of the target LLM on a sample to the zlib entropy of the same sample.

Min-K% Prob [31]. This heuristic computes the average of the minimum $K\%$ token probabilities outputted by the LLM on the sample. Note that this method requires tuning K, so in all our experiments we chose the best result over $K\% \in \{5\%, 10\%, 20\%, 30\%, 40\%, 50\%, 60\%\}$.

2.3 LLM Watermarking Methods

UMD [20] splits the vocabulary into two sets (green and red) and biases the model towards the green tokens by altering the logit distribution. The hash of the previous token's ID serves as a seed for a pseudo-random number generator used to split the vocabulary into these two groups. For a "hard" watermark, the model is forced not to sample from the red list at all. For a "soft" watermark, a positive bias δ is added to the logits of the green tokens before sampling. We focus our empirical evaluation on "soft" watermarks as they are more suitable for LLM deployment due to their smaller impact on the quality of the generated text.

Unigram-Watermark [43] employs a similar approach of splitting the vocabulary into two sets and biasing the model towards one of the two sets. However, the split remains consistent throughout the generation. This choice is made to provide a provable improvement against paraphrasing attacks [21].

3 Watermarking LLMs Prevents Copyrighted Text Generation

In this section, we study the effect of LLM watermarking techniques on verbatim memorization. We discuss the their implications for preventing copyrighted text generation.

Datasets. We consider 4 versions of the WikiMIA benchmark [31] with 32, 64, 128, and 256 words in each sample and only consider the samples that were very likely part of the training set of all the models we consider (labeled as 1 in Shi et al. [31]). We consider these subsets as a proxy for text that was used in the training set, and the model may be prone to verbatim memorization. From now on,

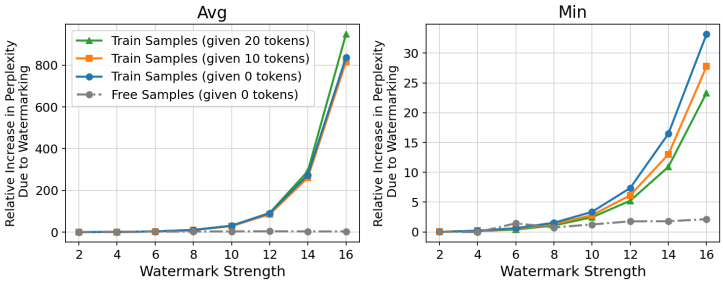

Figure 2: We study how the watermark strength (under the UMD scheme) affects the average and the minimum perplexity of training samples from WikiMIA-32, as well as the quality of generated text.

we refer to this subset as the "training samples" or "training texts". Similarly, we consider BookMIA dataset [31], which contains samples from copyrighted books.

Metric. We measure the relative increase in perplexity on the generation of training samples by the watermarked model compared to the original model. We report the increase in both the minimum and average perplexity over the training samples. Note that a large increase in perplexity corresponds to a large decrease in the probability of generating that specific sample, as shown later in this section. When computing the perplexity, we prompt the model with an empty string, the first 10, and the first 20 tokens of the targeted training sample, respectively. In the BookMIA dataset, we designate the initial 100 tokens as the prompt. This is because each BookMIA sample contains 512 words, which is larger than the sample size in WikiMIA.

Models. We conduct our empirical evaluation on 5 recent LLMs: Llama-30B [37], GPT-NeoX-20B [4], Llama-13B [37], Pythia-2.8B [3] and OPT-2.7B [42].

3.1 Empirical Evaluation

		Llama-30B		Llama-13B	
	P.	Min.	Avg.	Min.	Avg.
UMD	0	3.3	31.2	4.9	34.3
	10	2.8	28.7	3.5	31.9
	20	2.4	30.1	3.5	33.4
Unigram	0	4.1	34.1	5.0	36.6
	10	3.0	31.7	4.0	34.3
	20	2.4	31.5	3.4	34.0

Table 1: Measuring the reduction in verbatim memorization of training texts on WikiMIA-32. We report the relative increase in both the minimum and average perplexity between the watermarked and unwatermarked models, where larger values correspond to less memorization. Note that "P." stands for "prompt length".

In Table 1, we show the increase in perplexity on the training samples when the model is watermarked relative to the unwatermarked model. We observe that for Llama-30B, Unigram-Watermark induces a relative increase of 4.1 in the minimum and 34.1 in the average perplexity. Note that a relative increase of 4.1 in perplexity for a sample makes it more than 4.3×10^{22} times less likely to be generated. This is based on a sample with only 32 tokens, which is likely a lower bound since the number of tokens is typically larger than the number of words. We observe consistent results over several models and prompt lengths. For all experiments, unless otherwise specified, we use a fixed strength parameter $\delta = 10$ for watermark methods and a fixed percentage of 50% green tokens. All the results are averaged over 5 runs with different seeds for the watermark methods. We include additional results on WikiMIA-64, WikiMIA-128 and WikiMIA-256 in Tables 6, 7 and 8, respectively, in the appendix. We observe that our findings are consistent across models and splits of WikiMIA. Finally, we include the complete version of Table 1 in the appendix (Table 5), which shows results for additional models and random logit perturbations with the same strength as the watermarking methods. Overall, the additional results are consistent with our previous findings.

In Figure 2, as well as Figure 3 from the appendix, we study the influence of the strength of the watermark δ on the relative increase in both the minimum and average perplexity on the WikiMIA-32 training samples. In this experiment, we also consider a baseline of generating text freely to study the impact of watermarks on the quality of text relative to the impact on training samples' generation

(here, perplexity is computed by an unwatermarked model). All the results are averaged over 5 runs with different seeds for the watermark methods. In the case of free generation, we generate 100 samples for 5 different watermarking seeds and average the results. The length of the generated samples is up to 42 tokens, which is approximately 32 words in the benchmark (on a token-to-word ratio of $4:3$). **The results show an exponential increase in the perplexity of the training samples with the increase in watermark strength, while the generation quality is affected at a slower rate.** This suggests that even if there is a trade-off between protecting the generation of text memorized verbatim and generating high-quality text, **finding a suitable watermark strength for each particular application is possible**.

		Llama-30B		Llama-13B	
	P.	Min.	Avg.	Min.	Avg.
UMD	0	1.5	33.7	2.4	41.2
	10	1.5	33.6	2.3	41.0
	20	1.4	33.5	2.3	40.8
	100	1.3	32.9	1.9	40.3
Unigram	0	1.6	36.4	2.4	44.5
	10	1.6	36.3	2.4	44.3
	20	1.5	36.1	2.3	44.2
	100	1.4	35.5	1.8	43.6

Table 2: Measuring the reduction in verbatim memorization of training texts on BookMIA. We report the relative increase in both the minimum and average perplexity between the watermarked and unwatermarked models, where larger values correspond to less memorization. Note that "P." stands for "prompt length".

Takeaways. Watermarking significantly increases the perplexity of generating training texts, reducing verbatim memorization likelihood. This is achieved with only a moderate impact on the overall quality of generated text. This suggests that watermark strength can be effectively tailored to balance verbatim memorization and text quality for specific applications. Finally, we believe that our findings directly extend to the generation of copyrighted text verbatim, as this constitutes a form of verbatim memorization of the training data. Since copyrighted texts are not expected to be distributed significantly differently from the rest of the training data, the probability of generating copyrighted materials under watermarking is also likely to decrease. To confirm, we run similar experiments on a dataset containing copyrighted data (BookMIA) and include the results in the Table 2. Additionally, we consider finetuning Llama-7B [37] on BookMIA while controlling memorization by duplicating training samples. Detailed information about this experiment is provided in the appendix. Finally, we extend our results to *approximate memorization*, which we define and discuss in the appendix.

4 Impact of Watermarking on Pretraining Data Detection

Datasets. We revisit the WikiMIA benchmark as discussed in the previous section. We consider the full datasets, rather than the subset of samples that were part of the training for models we study. Additionally, we consider the BookMIA benchmark, which contains copyrighted texts.

Metrics. We follow the prior work [31, 13] and report the AUC and AUC drop to study the detection performance of the MIAs. Note that this metric has the advantage of not having to tune the threshold for the detection classifier.

Models. We conduct experiments on the same LLMs as in the previous section. Additionally, for the Smaller Ref method that requires a smaller reference model along with the target LLM, we consider Llama-7B, Neo-125M, Pythia-70M, and OPT-350M as references.

4.1 Empirical evaluation

In Table 4, we show the AUC for the unwatermarked and watermarked models using the UMD scheme, as well as the drop between the two. We observe that watermarking reduces the AUC (drop shown in bold in the table) by up to 14.2% across 4 detection methods and 5 LLMs. All the experiments on watermarked models are run with 5 different seeds and we report the mean and standard deviation of the results. We also report the AUC drop, which is computed by the difference between the AUC for the unwatermarked model and the mean AUC over the 5 runs for the watermarked model. Additionally, while the experiments from Table 4 are conducted on WikiMIA-256, we observe similar trends for WikiMIA-32, WikiMIA-64, and WikiMIA-128 in the appendix. We also study the impact of the watermark's strength on the AUC drop for Llama-30B in Figure 4 (from the appendix) and for

Table 4: AUC of each MIA for the unwatermarked (*top* of each cell), watermarked models (*middle* of each cell), and the drop between the two (*bottom* of each cell) on WikiMIA-256 using UMD scheme.

	Llama-30B	NeoX-20B	Llama-13B	Pythia-2.8B	OPT-2.7B
PPL	72.0% $70.6 \pm 1.9\%$ **1.4%**	71.3% $64.7 \pm 2.3\%$ **6.6%**	71.2% $70.0 \pm 2.6\%$ **1.2%**	67.8% $64.4 \pm 1.9\%$ **3.4%**	60.5% $54.9 \pm 2.2\%$ **5.6%**
Lowercase	68.1% $63.8 \pm 4.5\%$ **4.3%**	68.2% $55.4 \pm 5.5\%$ **14.2%**	65.5% $61.6 \pm 3.8\%$ **3.9%**	62.9% $58.7 \pm 3.2\%$ **4.2%**	58.9% $49.7 \pm 2.9\%$ **9.2%**
Zlib	72.7% $72.0 \pm 1.6\%$ **0.7%**	73.2% $66.6 \pm 2.0\%$ **6.6%**	73.1% $71.6 \pm 2.3\%$ **1.5%**	69.2% $66.1 \pm 1.2\%$ **3.1%**	62.7% $58.1 \pm 1.8\%$ **4.6%**
Min-K% Prob	71.8% $70.5 \pm 1.8\%$ **1.3%**	78.0% $76.2 \pm 2.1\%$ **1.8%**	72.9% $70.4 \pm 3.2\%$ **2.5%**	71.0% $69.5 \pm 1.6\%$ **1.5%**	65.5% $63.1 \pm 3.4\%$ **2.4%**

the other models in Figure 5 (from the appendix). Note that we considered WikiMIA-256 for these experiments. We observe that higher watermark strengths generally induce larger AUC drops.

	Llama-30B	Llama-13B
PPL	85.4% $84.7 \pm 1.4\%$ **0.7%**	68.2% $67.6 \pm 2.5\%$ **0.6%**
Lowercase	87.9% $80.9 \pm 3.1\%$ **7.0%**	77.6% $67.2 \pm 4.0\%$ **10.4%**
Zlib	82.5% $77.8 \pm 1.2\%$ **4.7%**	62.5% $57.1 \pm 2.0\%$ **5.4%**
Min-K% Prob	85.1% $85.0 \pm 1.0\%$ **0.1%**	70.2% $68.5 \pm 0.1\%$ **1.7%**

Table 3: AUC of each MIA for the unwatermarked (*top* of each cell), watermarked models (*middle* of each cell), and the drop between the two (*bottom* of each cell) on BookMIA using UMD scheme.

In addition to the 4 detection methods, we also consider Smaller Ref attack, which we include in Table 12 of the appendix. We consider different variations, including an unwatermarked reference model and a watermarked one with a similar strength but a different seed or with both strength and seed changed in comparison to the watermarked target model. The baseline is an unwatermarked model with an unwatermarked reference model. We observe the AUC drops in all scenarios (up to 16.4%), which is consistent with our previous findings.

We also experiment with several percentages of green tokens for a fixed watermark strength of $\delta = 10$. We show the results in Table 13 of the appendix. We observe that for all models, in at least 80% of the cases all of the attacks' AUCs are negatively affected (positive drop value), suggesting that, in general, **finding a watermarking scheme that reduces the success rates of the current MIAs is not a difficult task**. Note that the experiments are run on WikiMIA for UMD scheme and the results are averaged over 5 watermark seeds.

Takeaways. Watermarking can significantly reduce the success of membership inference attacks (MIAs), with AUC drops up to 16.4%. By varying the percentage of green tokens as well as the watermark's strength, we observe that watermarking schemes can be easily tuned to negatively impact the detection success rates of MIAs. Finally, we conduct experiments on the BookMIA dataset and observe results consistent with our previous findings. These results are included in Table 3.

5 Conclusion and Discussion

Watermarking LLMs has unintended consequences on methods towards copyright protection. Our experiments demonstrate that while watermarking may be a promising solution to prevent copyrighted text generation, watermarking also complicates membership inference attacks that may be employed to detect copyright abuses. Watermarking can be a double-edged sword for copyright regulators since it promotes compliance during generation time, while making training time copyright violations harder to detect. We hope our work further the discussion around watermarking and copyright issues for LLMs.

References

[1] S. Aaronson. Simons institute talk on watermarking of large language models. 2023. URL https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17.

[2] J. W. Bentley, D. Gibney, G. Hoppenworth, and S. K. Jha. Quantifying membership inference vulnerability via generalization gap and other model metrics. *arXiv preprint arXiv:2009.05669*, 2020.

[3] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[4] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

[5] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284, 2019.

[6] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[7] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[8] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

[9] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023.

[10] M. Christ, S. Gunn, and O. Zamir. Undetectable watermarks for language models. *arXiv preprint arXiv:2306.09194*, 2023.

[11] T. Chu, Z. Song, and C. Yang. How to protect copyright data in optimization of large language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17871–17879, 2024.

[12] D. Das, J. Zhang, and F. Tramèr. Blind baselines beat membership inference attacks for foundation models. *arXiv preprint arXiv:2406.16201*, 2024.

[13] A. V. Duarte, X. Zhao, A. L. Oliveira, and L. Li. De-cop: Detecting copyrighted content in language models training data. *arXiv preprint arXiv:2402.09910*, 2024.

[14] N. Elkin-Koren, U. Hacohen, R. Livni, and S. Moran. Can copyright be reduced to privacy? *arXiv preprint arXiv:2305.14822*, 2023.

[15] U. Hacohen, A. Haviv, S. Sarfaty, B. Friedman, N. Elkin-Koren, R. Livni, and A. H. Bermano. Not all similarities are created equal: Leveraging data-driven biases to inform genai copyright disputes. *arXiv preprint arXiv:2403.17691*, 2024.

[16] A. Hans, Y. Wen, N. Jain, J. Kirchenbauer, H. Kazemi, P. Singhania, S. Singh, G. Somepalli, J. Geiping, A. Bhatele, et al. Be like a goldfish, don't memorize! mitigating memorization in generative llms. *arXiv preprint arXiv:2406.10209*, 2024.

[17] D. Ippolito, F. Tramèr, M. Nasr, C. Zhang, M. Jagielski, K. Lee, C. A. Choquette-Choo, and N. Carlini. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53. Association for Computational Linguistics, 2023.

[18] N. Kandpal, E. Wallace, and C. Raffel. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR, 2022.

[19] A. Karamolegkou, J. Li, L. Zhou, and A. Søgaard. Copyright violations and large language models. *arXiv preprint arXiv:2310.13771*, 2023.

[20] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR, 2023.

[21] K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] R. Kuditipudi, J. Thickstun, T. Hashimoto, and P. Liang. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

[23] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.

[24] A. Liu, L. Pan, X. Hu, S. Meng, and L. Wen. A semantic invariant robust watermark for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=6p8lpe4MNf`.

[25] J. Mattern, F. Mireshghallah, Z. Jin, B. Schoelkopf, M. Sachan, and T. Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.719. URL `https://aclanthology.org/2023.findings-acl.719`.

[26] F. Mireshghallah, K. Goyal, A. Uniyal, T. Berg-Kirkpatrick, and R. Shokri. Quantifying privacy risks of masked language models using membership inference attacks. *arXiv preprint arXiv:2203.03929*, 2022.

[27] Y. Oren, N. Meister, N. Chatterji, F. Ladhak, and T. B. Hashimoto. Proving test set contamination in black box language models. *arXiv preprint arXiv:2310.17623*, 2023.

[28] J. Ren, H. Xu, P. He, Y. Cui, S. Zeng, J. Zhang, H. Wen, J. Ding, H. Liu, Y. Chang, et al. Copyright protection in generative ai: A technical perspective. *arXiv preprint arXiv:2402.02333*, 2024.

[29] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR, 2019.

[30] V. Shejwalkar, H. A. Inan, A. Houmansadr, and R. Sim. Membership inference attacks against nlp classification models. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[31] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 2023.

[32] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[33] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023.

[34] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023.

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

[36] K. Tirumala, A. Markosyan, L. Zettlemoyer, and A. Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

[37] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[38] N. Vyas, S. M. Kakade, and B. Barak. On provable copyright protection for generative models. In *International Conference on Machine Learning*, pages 35277–35299. PMLR, 2023.

[39] Y. Wen, Y. Liu, C. Chen, and L. Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=84n3UwkH7b`.

[40] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[41] J. Zhang, J. Sun, E. Yeats, Y. Ouyang, M. Kuo, J. Zhang, H. Yang, and H. Li. Min-k%++: Improved baseline for detecting pre-training data from large language models. *arXiv preprint arXiv:2404.02936*, 2024.

[42] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[43] X. Zhao, P. Ananth, L. Li, and Y.-X. Wang. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*, 2023.

A Appendix

A.1 Additional experiments on verbatim memorization on WikiMIA

Table 5: Measuring the reduction in verbatim memorization of training texts on WikiMIA-32. We report the relative increase in both the minimum and average perplexity between the watermarked and unwatermarked models, where larger values correspond to less memorization. Note that "P." stands for "prompt length".

		Llama-30B		NeoX-20B		Llama-13B		Pythia-2.8B		OPT-2.7B	
	P.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.
UMD	0	3.3	31.2	3.7	52.1	4.9	34.3	11.4	61.3	10.4	64.5
	10	2.8	28.7	2.2	52.1	3.5	31.9	8.8	63.7	8.3	67.7
	20	2.4	30.1	1.8	66.0	3.5	33.4	5.0	74.0	7.0	84.4
Unigram	0	4.1	34.1	4.4	54.1	5.0	36.6	14.3	74.5	11.5	66.1
	10	3.0	31.7	2.8	52.5	4.0	34.3	11.8	73.6	9.8	70.2
	20	2.4	31.5	2.0	56.4	3.4	34.0	6.6	79.1	5.8	81.4
Random	0	4.0	34.3	4.9	51.1	5.5	34.7	8.0	62.3	7.4	60.6
	10	2.6	31.4	3.1	51.4	3.6	31.7	5.5	62.9	6.3	64.0
	20	2.1	31.8	1.2	59.3	2.8	32.9	4.7	78.6	3.9	73.7

Table 6: Measuring the reduction in verbatim memorization of training texts on WikiMIA-64. We report the relative increase in both the minimum and average perplexity between the watermarked and unwatermarked models, where larger values correspond to less memorization. Note that "P." stands for "prompt length".

		Llama-30B		NeoX-20B		Llama-13B		Pythia-2.8B		OPT-2.7B	
	P.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.
UMD	0	4.9	27.6	4.2	42.8	6.7	30.5	15.2	50.4	14.9	51.7
	10	4.3	26.2	3.7	41.3	6.1	29.1	15.6	49.2	14.4	51.4
	20	3.9	26.4	3.6	43.1	5.8	29.3	14.0	50.3	12.5	52.7
Unigram	0	5.0	28.1	4.3	45.3	6.7	30.9	17.6	62.3	16.0	53.7
	10	3.8	26.9	3.4	43.6	5.3	29.7	16.2	60.6	17.1	53.0
	20	3.2	26.9	3.1	44.2	4.4	29.7	13.6	60.9	11.7	53.6

Table 7: Measuring the reduction in verbatim memorization of training texts on WikiMIA-128. We report the relative increase in both the minimum and average perplexity between the watermarked and unwatermarked models, where larger values correspond to less memorization. Note that "P." stands for "prompt length".

		Llama-30B		NeoX-20B		Llama-13B		Pythia-2.8B		OPT-2.7B	
	P.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.
UMD	0	5.7	25.3	4.6	39.5	7.6	28.0	23.1	45.3	18.6	48.1
	10	5.3	24.4	4.3	38.9	7.2	27.1	23.6	44.7	19.1	47.6
	20	5.2	24.5	4.3	39.3	6.8	27.2	23.0	44.7	17.5	47.8
Unigram	0	4.5	25.6	5.9	42.9	6.4	28.2	17.6	54.9	19.6	50.0
	10	3.9	25.0	5.3	42.0	5.7	27.6	15.8	53.6	18.7	49.9
	20	3.6	25.2	5.1	42.1	5.3	27.7	15.1	53.6	18.0	49.9

Table 8: Measuring the reduction in verbatim memorization of training texts on WikiMIA-256. We report the relative increase in both the minimum and average perplexity between the watermarked and unwatermarked models, where larger values correspond to less memorization. Note that "P." stands for "prompt length".

		Llama-30B		NeoX-20B		Llama-13B		Pythia-2.8B		OPT-2.7B	
	P.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.
UMD	0	7.5	23.9	15.4	37.8	13.0	26.3	31.2	45.4	27.3	46.3
	10	7.3	23.4	15.5	37.6	12.5	25.8	30.9	45.2	27.5	46.0
	20	7.2	23.5	16.1	37.6	12.6	25.9	30.4	45.0	27.6	46.2
Unigram	0	7.4	24.4	21.0	42.4	13.9	26.8	36.9	54.3	28.8	46.3
	10	7.1	24.1	21.2	41.9	13.7	26.5	35.4	53.7	28.2	46.0
	20	6.7	24.2	21.5	41.8	13.7	26.5	34.6	53.4	29.3	45.9

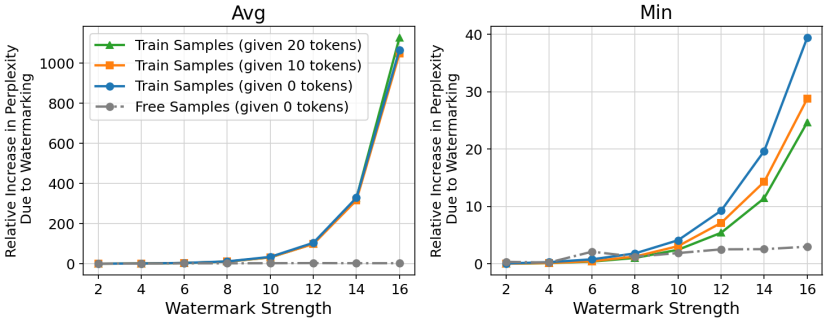

Figure 3: We study how the watermark strength (under the Unigram scheme) affects the average and the minimum perplexity of training samples from WikiMIA-32, as well as the quality of generated text.

A.2 Additional experiments on pretraining data detection on WikiMIA

Table 9: AUC of each MIA for the unwatermarked (*top* of each cell), watermarked models (*middle* of each cell) and the drop between the two (*bottom* of each cell) on WikiMIA-128 using UMD scheme.

	Llama-30B	NeoX-20B	Llama-13B	Pythia-2.8B	OPT-2.7B
PPL	70.3% $66.3 \pm 2.2\%$ **4.0%**	70.6% $63.6 \pm 2.4\%$ **7.0%**	67.7% $63.4 \pm 2.6\%$ **4.3%**	62.8% $61.4 \pm 2.3\%$ **1.4%**	60.0% $55.1 \pm 1.6\%$ **4.9%**
Lowercase	59.1% $55.9 \pm 2.9\%$ **3.2%**	68.0% $58.2 \pm 3.4\%$ **9.2%**	60.6% $55.1 \pm 3.0\%$ **5.5%**	59.4% $55.7 \pm 1.6\%$ **3.7%**	57.1% $49.2 \pm 4.5\%$ **7.9%**
Zlib	71.8% $68.6 \pm 2.3\%$ **3.2%**	72.3% $66.3 \pm 2.1\%$ **6.0%**	69.6% $65.8 \pm 2.7\%$ **3.8%**	64.9% $63.9 \pm 1.9\%$ **1.0%**	62.3% $58.9 \pm 1.3\%$ **3.4%**
Min-K% Prob	73.8% $70.0 \pm 1.5\%$ **3.8%**	76.4% $72.8 \pm 2.3\%$ **3.6%**	71.5% $68.9 \pm 2.2\%$ **2.6%**	66.8% $64.8 \pm 1.4\%$ **2.0%**	64.3% $59.2 \pm 2.4\%$ **5.1%**

Table 10: AUC of each MIA for the unwatermarked (*top* of each cell), watermarked models (*middle* of each cell) and the drop between the two (*bottom* of each cell) on WikiMIA-64 using UMD scheme.

	Llama-30B	NeoX-20B	Llama-13B	Pythia-2.8B	OPT-2.7B
PPL	66.1% $60.7 \pm 3.4\%$ **5.4%**	66.6% $60.1 \pm 3.2\%$ **6.5%**	63.6% $58.0 \pm 3.7\%$ **5.6%**	58.4% $58.7 \pm 1.7\%$ **-0.3%**	55.1% $52.2 \pm 2.1\%$ **2.9%**
Lowercase	61.8% $54.8 \pm 1.7\%$ **7.0%**	66.4% $56.8 \pm 3.8\%$ **9.6%**	62.0% $53.8 \pm 1.1\%$ **8.2%**	57.7% $54.5 \pm 1.0\%$ **3.2%**	56.6% $51.4 \pm 3.1\%$ **5.2%**
Zlib	67.4% $62.4 \pm 3.3\%$ **5.0%**	68.1% $62.0 \pm 2.6\%$ **6.1%**	65.3% $59.9 \pm 3.6\%$ **4.9%**	60.5% $60.9 \pm 1.8\%$ **5.4%**	57.7% $55.5 \pm 1.5\%$ **2.2%**
Min-K% Prob	68.4% $64.4 \pm 2.9\%$ **4.0%**	72.8% $67.7 \pm 3.3\%$ **5.1%**	65.9% $62.8 \pm 3.4\%$ **3.1%**	61.2% $59.8 \pm 0.7\%$ **1.4%**	58.0% $55.3 \pm 2.3\%$ **2.7%**

Table 11: AUC of each MIA for the unwatermarked (*top* of each cell), watermarked models (*middle* of each cell) and the drop between the two (*bottom* of each cell) on WikiMIA-32 using UMD scheme.

	Llama-30B	NeoX-20B	Llama-13B	Pythia-2.8B	OPT-2.7B
PPL	69.4% $63.6 \pm 5.2\%$ **5.5%**	69.0% $62.7 \pm 3.5\%$ **6.3%**	67.5% $61.4 \pm 5.7\%$ **6.1%**	61.3% $60.8 \pm 2.3\%$ **0.5%**	58.2% $55.2 \pm 2.1\%$ **3.0%**
Lowercase	64.1% $54.9 \pm 1.8\%$ **9.2%**	68.2% $59.4 \pm 4.8\%$ **8.8%**	63.9% $54.2 \pm 1.8\%$ **9.7%**	60.9% $55.5 \pm 1.6\%$ **0.6%**	59.2% $52.1 \pm 3.9\%$ **2.8%**
Zlib	69.8% $64.4 \pm 4.7\%$ **5.4%**	69.2% $63.2 \pm 2.8\%$ **6.0%**	67.8% $62.3 \pm 5.1\%$ **5.5%**	62.1% $61.5 \pm 1.9\%$ **0.6%**	59.4% $56.6 \pm 1.6\%$ **2.8%**
Min-K% Prob	70.1% $66.2 \pm 4.2\%$ **3.9%**	72.1% $67.1 \pm 4.2\%$ **5.0%**	67.9% $64.5 \pm 4.1\%$ **3.4%**	61.8% $61.0 \pm 1.5\%$ **0.8%**	59.2% $55.8 \pm 2.3\%$ **3.4%**

Table 12: Results for Smaller Ref attack on WikiMIA-256. The first two rows represent the pair of target and smaller reference model, "No model w." row represents the baseline AUC of a unwatermarked target LLM and unwatermarked reference model, the other three "double rows" correspond to different variations of the reference model and each cell contains the AUC followed by the AUC drop in comparison to the baseline.

	Llama-30B	NeoX-20B	Llama-13B	Pythia-2.8B	OPT-2.7B
	Llama-7B	Neo-125M	Llama-7B	Pythia-70M	OPT-350M
No model w.	74.7%	70.2%	70.5%	63.6%	64.4%
Ref. not w.	$69.7 \pm 3.3\%$ **5.0%**	$61.0 \pm 1.8\%$ **9.2%**	$66.3 \pm 4.6\%$ **4.2%**	$61.6 \pm 2.0\%$ **2.0%**	$53.2 \pm 3.4\%$ **11.2%**
Ref. diff. seed	$61.7 \pm 4.4\%$ **13.0%**	$55.5 \pm 3.4\%$ **15.0%**	$54.1 \pm 4.4\%$ **16.4%**	$58.3 \pm 2.4\%$ **5.3%**	$51.3 \pm 4.3\%$ **13.1%**
Ref. diff. str.	$73.7 \pm 2.6\%$ **1.0%**	$61.0 \pm 3.2\%$ **9.2%**	$68.8 \pm 4.8\%$ **1.7%**	$62.5 \pm 1.2\%$ **1.1%**	$57.3 \pm 3.6\%$ **7.1%**

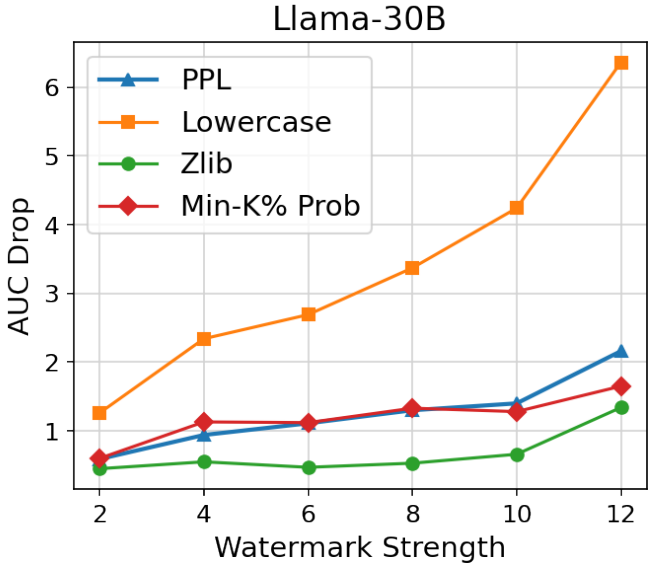

Figure 4: AUC drop due to watermarking for each MIA when varying the strength of the watermark.

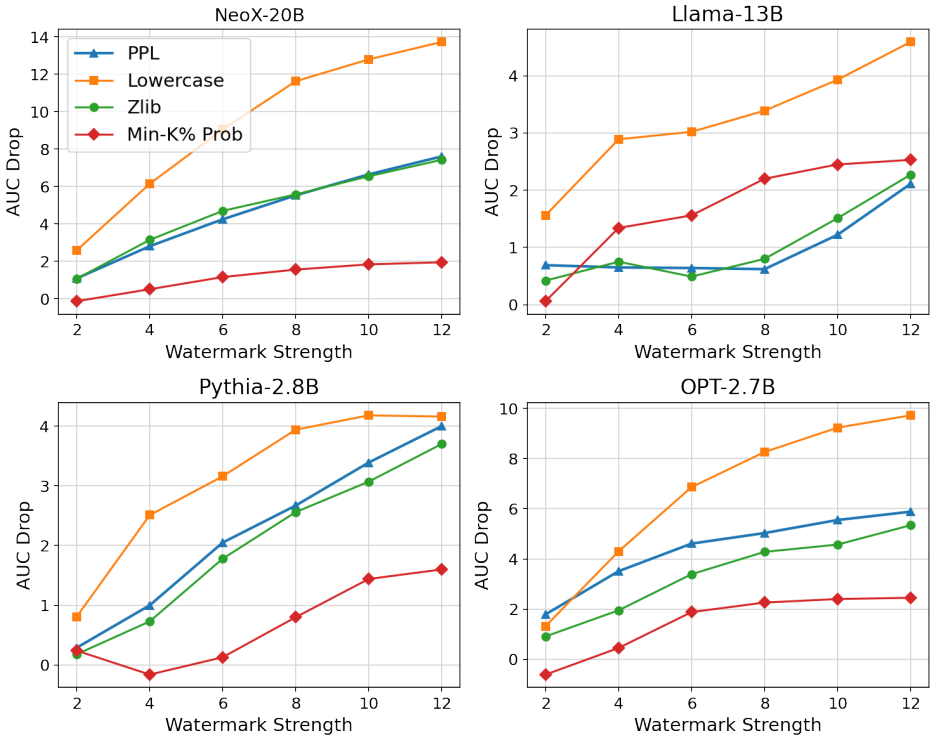

Figure 5: AUC drop due to watermarking for each MIA when varying the strength of the watermark.

Table 13: We show the AUC drop when we vary the percentage of green tokens between 30% and 70%. We bold the scenarios when a specific percentage value induces AUC drops for all the attacks.

		0.3	0.4	0.5	0.6	0.7
Llama-30B	PPL	**0.71**	-0.10	**1.40**	**2.14**	**1.65**
	Lowercase	**0.27**	2.60	**4.24**	**2.45**	**3.15**
	Zlib	**0.58**	-0.18	**0.66**	**1.10**	**0.52**
	Min-K% Prob	**1.38**	0.03	**1.28**	**1.64**	**1.45**
NeoX-20B	PPL	**5.56**	**5.65**	**6.64**	**6.92**	**4.96**
	Lowercase	**9.84**	**11.12**	**12.79**	**11.94**	**9.78**
	Zlib	**6.68**	**5.94**	**6.54**	**6.51**	**4.71**
	Min-K% Prob	**0.45**	**0.88**	**1.83**	**1.26**	**3.84**
Llama-13B	PPL	**0.19**	-0.86	**1.22**	**1.84**	**1.39**
	Lowercase	**0.49**	2.45	**3.93**	**1.54**	**1.65**
	Zlib	**1.31**	0.28	**1.51**	**2.00**	**1.29**
	Min-K% Prob	**2.81**	1.21	**2.45**	**2.70**	**2.78**
Pythia-2.8B	PPL	**4.65**	**4.49**	**3.39**	**4.42**	**3.66**
	Lowercase	**4.91**	**6.23**	**4.18**	**5.33**	**7.22**
	Zlib	**5.37**	**3.97**	**3.07**	**3.17**	**2.20**
	Min-K% Prob	**1.10**	**1.21**	**1.44**	**3.42**	**4.71**
OPT-2.7B	PPL	**5.45**	**5.39**	**5.55**	**5.18**	**5.76**
	Lowercase	**7.71**	**9.91**	**9.23**	**9.83**	**7.28**
	Zlib	**3.35**	**4.12**	**4.57**	**4.17**	**4.11**
	Min-K% Prob	**2.30**	**2.07**	**2.40**	**3.90**	**5.00**

A.3 Additional experiments on BookMIA

In this section, we conduct experiments using models finetuned on a subset of BookMIA, which we refer to as BookMIA-2. To build BookMIA-2, we first select only the samples that were not part of the training set of any model that we consider (labeled as 0 by [31]). Then, we randomly select half of them as finetuning data (referred to as *seen* samples) and keep the other half as *unseen* samples. Note that for BookMIA-2, there would not be a distribution difference between the seen and unseen samples [12]. We also consider duplicating a sample from the training set of BookMIA-2 to have more fine-grained control over the memorization of that sample. We use Llama-7B in all the experiments from this section.

Verbatim Memorization. We study verbatim memorization on BookMIA-2 by measuring the relative increase in perplexity on the generation of the duplicated sample by the watermarked model compared to the original model, as well as the ratio between the probability of generating the duplicated sample by the original model to the watermarked model, which we refer to as probability reduction factor. We run each experiment with 20 seeds and report the average perplexity and the minimum probability reduction factor. We consider both the UMD and Unigram watermarking methods with several strengths (2, 5, and 10) and prompt the model with an empty string, as well as with the first 10, 20, and 100 words from the training sample. Additionally, we consider several duplication factors (the number of times one randomly chosen target sample appears in the dataset): 1, 10, 20, and 50. We show the results in Table 14. We observe that even in high memorization cases (duplication factor of 50), as long as the watermark is strong enough, the probability of generating the memorized sample decreases by almost 200 orders of magnitude, making it very unlikely to be generated.

Table 14: Average relative increase in perplexity and minimum probability reduction factor for generating the memorized target sample from BookMIA-2. Note that "S.", "P.", and "D." stand for watermark method's strength, prompt length, and duplication factor, respectively.

			D = 1		D = 10		D = 20		D = 50	
	S.	P.	PPL.	Prob.	PPL.	Prob.	PPL.	Prob.	PPL.	Prob.
UMD	2	0	0.32	3.1×10^{70}	0.28	2.9×10^{45}	0.17	1.1×10^{19}	0.007	4.4×10^{0}
		10	0.32	1.9×10^{69}	0.28	1.2×10^{44}	0.17	1.3×10^{18}	0.005	5.1×10^{0}
		20	0.32	1.7×10^{67}	0.28	2.9×10^{42}	0.17	4.8×10^{17}	0.005	5.0×10^{0}
		100	0.32	3.7×10^{57}	0.28	1.6×10^{34}	0.16	3.3×10^{10}	0.005	4.2×10^{0}
	5	0	2.93	1.1×10^{366}	2.58	1.8×10^{261}	1.53	1.6×10^{122}	0.10	1.8×10^{18}
		10	2.92	7.6×10^{360}	2.56	6.9×10^{257}	1.51	8.9×10^{99}	0.09	1.1×10^{16}
		20	2.91	1.2×10^{354}	2.55	2.6×10^{254}	1.50	3.0×10^{98}	0.09	4.3×10^{15}
		100	2.89	4.9×10^{313}	2.53	4.2×10^{213}	1.45	6.0×10^{71}	0.09	6.7×10^{13}
	10	0	38.6	3.2×10^{1028}	33.0	6.7×10^{883}	18.6	1.9×10^{508}	1.86	7.1×10^{245}
		10	38.5	1.1×10^{1006}	32.8	9.2×10^{869}	18.4	2.5×10^{495}	1.78	4.7×10^{226}
		20	38.4	2.4×10^{982}	32.7	3.7×10^{851}	18.2	3.8×10^{482}	1.76	4.6×10^{223}
		100	38.0	2.8×10^{860}	32.3	6.6×10^{731}	17.7	1.0×10^{388}	1.70	3.8×10^{199}
Unigram	2	0	0.32	5.2×10^{63}	0.29	6.8×10^{59}	0.17	1.2×10^{14}	0.008	2.9×10^{0}
		10	0.32	4.7×10^{62}	0.29	3.6×10^{58}	0.17	4.8×10^{14}	0.005	5.9×10^{0}
		20	0.32	7.9×10^{61}	0.29	9.8×10^{56}	0.17	3.1×10^{14}	0.005	5.8×10^{0}
		100	0.31	1.2×10^{50}	0.28	5.2×10^{46}	0.16	2.7×10^{7}	0.005	5.1×10^{0}
	5	0	2.88	3.4×10^{304}	2.56	1.2×10^{290}	1.53	1.1×10^{79}	0.11	2.2×10^{22}
		10	2.87	1.6×10^{300}	2.56	1.9×10^{286}	1.52	8.6×10^{77}	0.10	5.9×10^{16}
		20	2.87	8.2×10^{295}	2.55	3.1×10^{282}	1.50	6.2×10^{76}	0.09	4.6×10^{16}
		100	2.82	2.5×10^{261}	2.50	6.6×10^{239}	1.45	1.0×10^{47}	0.09	4.1×10^{14}
	10	0	37.6	2.4×10^{834}	32.3	2.2×10^{811}	18.5	1.8×10^{425}	1.87	2.6×10^{268}
		10	37.7	3.5×10^{824}	32.3	7.1×10^{799}	18.3	1.3×10^{419}	1.80	5.7×10^{251}
		20	37.6	4.9×10^{812}	32.2	2.3×10^{788}	18.1	6.1×10^{405}	1.78	1.3×10^{248}
		100	36.9	2.4×10^{707}	31.5	9.5×10^{684}	17.4	1.5×10^{323}	1.72	1.0×10^{212}

Approximate Memorization. Informally, we consider a training sample approximately memorized by a model if, given its prefix, it is possible to generate a completion that is similar enough to the ground truth completion. In our experiments, we consider Normalized Edit Similarity (referred to as edit similarity from now on) and BLEU score as similarity measures, as in [17]. Note that we consider both word-level and token-level variants for the BLEU score. The range for each metric is between 0 and 1, where values close to 1 represent similar texts. In all experiments, since all

the samples are 512 words long, we consider the first 256 words as the prefix and the last 256 words as the ground truth completion. We show the results in Figure 6 averaged over 20 runs with different seeds. Note that the duplication factor (shown on x-axis) represents the number of times the target copyrighted text is duplicated. We observe that for high levels of memorization, a strong watermark significantly reduces the similarity between the generated completion and the ground truth (copyrighted) one.

MIA. We also study the effect of the watermark on the effectiveness of MIAs for copyrighted training data detection (on BookMIA-2, without duplicated samples). We show the results in Table 15 and observe that watermarking negatively affects MIAs' success rate, which is consistent with our previous findings (from Section 4). Finally, we also run our adaptive method from Section C and observe an improvement of 0.9% over Min-K% Prob.

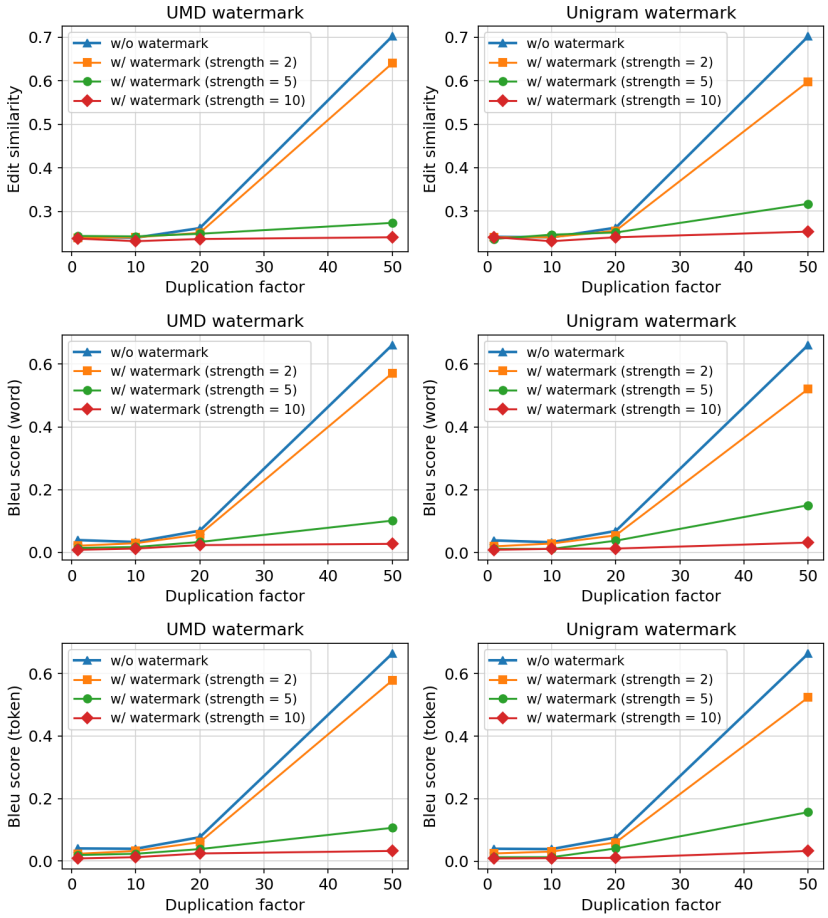

Figure 6: Edit similarity (*top*), word-level BLEU score (*middle*), and token-level BLEU score (*bottom*) between the generated completion and the ground truth when considering different watermark strengths on BookMIA-2.

A.4 Computing Infrastructure

All of our experiments were run on either three Nvidia RTX A6000 or four Nvidia RTX A5000 GPUs, using 128 GB of memory. We used the Transformers library (version 4.35.2) and PyTorch (version 2.1.0).

Table 15: AUC of each MIA for the unwatermarked (*top* of each cell), watermarked models (*middle* of each cell), and the drop between the two (*bottom* of each cell) on BookMIA-2 (without any duplicated samples) using the UMD scheme with a strength of 10. We average the results over 5 runs with different seeds.

	Llama-7B (fine-tuned)
PPL	$58.5 \pm 0.0\%$ $56.6 \pm 0.0\%$ **1.9%**
Lowercase	$59.8 \pm 0.1\%$ $52.9 \pm 0.3\%$ **6.9%**
Zlib	$59.7 \pm 0.0\%$ $56.1 \pm 0.1\%$ **3.6%**
Min-K% Prob	$58.5 \pm 0.0\%$ $57.1 \pm 0.2\%$ **1.4%**

A.5 Limitations & Future Work

Our work considers only decoding time watermarking techniques, future work may benefit from studying other types of watermarking methods. Our proposed method for improving MIAs' success rate on watermarked models makes strong assumptions on the watermarking scheme, which may not always be satisfied despite empirical improvements in our experiments. Our observations on the deterioration of MIAs' success suggests that for copyright violation auditing, an unwatermarked model or the watermarking scheme may be needed. We encourage the community to further refine adaptive methods to ensure robust copyright protection and data privacy, and consider the interactions of different methods on downstream legal concerns.

B Related work

Watermarks for LLMs. Language model watermarking techniques embed identifiable markers into output text to detect AI-generated content. Recent strategies incorporate watermarks during the decoding phase of language models [43, 20]. Aaronson [1] develops the Gumbel watermark, which employs traceable pseudo-random sampling for generating subsequent tokens. Kirchenbauer et al. [20] splits the vocabulary into red and green lists according to preceding tokens, biasing the generation towards green tokens. Zhao et al. [43] employs a fixed grouping strategy to develop a robust watermark with theoretical guarantees. Liu et al. [24] proposes to generate watermark logits based on the preceding tokens' semantics rather than their token IDs to boost the robustness. Kuditipudi et al. [22] and Christ et al. [10] explore watermark methods that do not change the output textual distribution.

Copyright. Copyright protection in the age of AI has gained importance, as discussed by Ren et al. [28]. Vyas et al. [38] addresses content protection through near access-freeness (NAF) and developed learning algorithms for generative models to ensure compliance under NAF conditions. Prior works focus on training algorithms to prevent copyrighted text generation [38, 11], whereas our work emphasizes lightweight, inference-time algorithms. Other works have studied copyright in machine learning from a legal perspective. Hacohen et al. [15] utilizes a generative model to determine the generic characteristics of works to aid in defining the scope of copyright. Elkin-Koren et al. [14] demonstrates that copying does not necessarily constitute copyright infringement and argues that existing detection methods may detract from the foundational purposes of copyright law.

Memorization. One cause of copyright issues is that machine learning models may memorize training data. Prior studies have observed that LLMs can memorize private information in training data, such as phone numbers and addresses [19, 5, 6, 23], leading to significant privacy

and security concerns. To measure memorization, Carlini et al. [6] proposes eidetic memorization, defining a string as memorized if it was present in the training data and it can be reproduced by a prompt. This definition, along with variations like exact and perfect memorization, has been widely adopted in subsequent studies [36, 18]. Carlini et al. [8] quantitatively measures memorization in LLMs as the fraction of extractable training data and finds that memorization significantly grows as model size scales and training examples are duplicated. To minimize memorization, Lee et al. [23] and Kandpal et al. [18] propose deduplicating training data, which also improves accuracy. Hans et al. [16] proposes the Goldfish Loss as a training-time defense against verbatim memorization. Ippolito et al. [17] proposes an inference time defense that perfectly prevents all verbatim memorization. However, it cannot prevent the leakage of training data due to the existence of many "style-transfer" prompts, suggesting it is a challenging open problem. Unlike the methods that we are studying in this paper, Ippolito et al. [17] requires access to a complete set of copyrighted texts that the model was trained on. Memorization in the image domain has also been studied from various angles [33, 34, 9, 39].

Membership Inference. As a proxy for measuring memorization, membership inference attacks (MIAs) predict whether or not a particular example was used to train the model [32, 40, 2]. Most membership inference attacks rely only on the model's loss since the model is more likely to overfit an example if it is in the training data [29]. Carlini et al. [7] trains shadow models to predict whether an example is from the training data. In the NLP domain, many works have focused on masked language models [26] and fine-tuning data detection [35, 30]. Recently, Shi et al. [31] studies pretraining data inference and introduced a detection method based on the hypothesis that unseen examples are likely to contain outlier words with low probabilities under the LLM. Zhang et al. [41] approaches pretraining data detection by measuring how sharply peaked the likelihood is around the inputs. Duarte et al. [13] proposes detecting copyrighted content in training data by probing the LLM with multiple-choice questions, whose options include both verbatim text and their paraphrases. Other methods include testing perplexity differences [25] and providing provable guarantees of test set contamination without access to pretraining data or model weights [27].

C Improving Detection Performance with Adaptive Min-K% Prob

This section demonstrates how an informed, adaptive attacker can improve the success rate of a recent MIA, Min-K% Prob. Our main idea is that an attacker with knowledge of the watermarking technique (including green-red token lists and watermark's strength δ) can readjust token probabilities. This is possible even without additional information about the logit distribution, relying solely on the probability of each token from the target sample given the preceding tokens. Our approach relies on two key assumptions. First, knowledge of the watermarking scheme, which aligns with assumptions made in prior work on public watermark detection [20]. Second, access to the probability of each token in a sample, given the previous tokens—an assumption also made by the Min-K% Prob method [31].

Our method described in Algorithm 1 is based on the observation that if the denominator of softmax function (i.e., $\sum_i e^{z_i}$, where z_i is the logit for the i-th vocabulary) does not vary significantly when generating samples with the watermarked model (and similarly for the unwatermarked model), then we can readjust the probabilities of the green tokens by "removing" the bias δ. More precisely, assuming the approximation for the denominator of softmax is good, then the probability for each token t_i in an unwatermarked model will be around $\frac{e^{L_i}}{c}$, where L_i is the logit corresponding to the token t_i and c is a constant. However, for a watermarked model, if the token t_i is green, then the probability would be approximated by $\frac{e^{L_i+\delta}}{d}$, where d is again a constant, while in the case t_i is red the probability will be around $\frac{e^{L_i}}{d}$. To compensate for the bias introduced by watermarking, we divide the probability of green tokens by e^{δ} and this way we end up with probabilities that are just a scaled (by $\frac{c}{d}$) version of the probabilities from the unwatermarked model. The scaling factor will not affect the orders between the samples when computing the average of the minimum K% log-probabilities as long as the tested sentences are approximately the same length, which is an assumption made by Shi et al. [31] as well.

Despite the strong assumption we assumed regarding the approximation of the denominator, empirical results show that our method effectively improves the success rate of Min-K% under

watermarking. We show results in Table 16. We observe that our method improves over the baseline in 95% of the cases, and the increase is as high as 4.8% (averaged over 5 runs).

Takeaways. We demonstrate that an adaptive attacker can leverage the knowledge of a watermarking scheme to increase the success rate of a recent MIA, Min-K% Prob.

Table 16: We show the AUC of Min-%K Prob (referred as "Not adapt.") and our method (referred as "Adapt.") when using UMD watermarking scheme. We highlight the cases when our method improves over the baseline.

		Llama-30B	NeoX-20B	Llama-13B	Pythia-2.8B	OPT-2.7B
WikiMIA 32	Not adapt.	66.2%	67.1%	64.5%	61.0%	55.7%
	Adapt.	**68.5%**	**71.3%**	**66.3%**	61.0%	**59.1%**
WikiMIA 64	Not adapt.	64.4%	67.7%	62.8%	59.8%	55.3%
	Adapt.	**67.3%**	**72.0%**	**64.9%**	**60.6%**	**57.4%**
WikiMIA 128	Not adapt.	70.0%	73.0%	68.9%	64.8%	59.2%
	Adapt.	**73.1%**	**75.9%**	**71.0%**	**66.4%**	**64.0%**
WikiMIA 256	Not adapt.	70.5%	76.2%	70.4%	69.5%	63.1%
	Adapt.	**71.3%**	**78.2%**	**72.4%**	**70.7%**	**66.2%**

Algorithm 1: Adaptive Min-K% Prob

Require : Tokenized target sample $t = t_1 \oplus t_2 \oplus ... \oplus t_n$, access to the probability of the target (watermarked) LLM f to generate t_i given the $i-1$ previous tokens and t_0 (empty string) $f(t_i|t_0 \oplus t_1 \oplus ... \oplus t_{i-1})$ (similar assumption as Min-K% Prob algorithm), K, we assume we know the watermarking scheme (e.g., for public watermark detection purposes), i.e. we know the green and red lists as well as δ.

Output : Adjusted average of the minimum $K\%$ token probabilities when generating $t_1 \oplus t_2 \oplus ... \oplus t_n$

```
adj_prob ← {}                                  ▷ The set of adjusted probabilities
for i ∈ 1, 2, ..., n do
    p_f(t_i) ← f(t_i | t_0 ⊕ t_1 ⊕ ... ⊕ t_{i-1})
    if t_i is green then
        adj_prob ← adj_prob ∪ {p_f(t_i) / e^δ}
    else
        adj_prob ← adj_prob ∪ {p_f(t_i)}
    end
end
k = floor(n · K%)        ▷ Find the number of token probabilities to keep
adj_k_prob ← min_k(adj_prob)        ▷ Select the minimum k probabilities
return mean(log(adj_k_prob) )        ▷ Return the mean of the minimum k
  log-probabilities
```

D Theoretical analysis

Notations and assumptions. We assume that the set of all copyrighted texts C_D that were part of the training data has m elements $\{s_1, s_2, ..., s_m\}$. Also, we assume that each copyrighted text has a fixed length n, and they are independent from each other.

Theorem 1 *For an LLM watermarked using a "hard" UMD scheme with a percentage of γ green tokens, then the probability of generating a copyrighted text is lower than $m \cdot \gamma^n$.*

Proof. Given one sample $s = t_1 \oplus t_2 \oplus ... \oplus t_n \in C_D$. For a "hard" watermarking scheme, the probability $P(s)$ of generating s is smaller than the probability of each token t_i to be on a green list. So, $P(s) < \gamma^n$. The probability of not generating any $s_j \in C_D$ is $P(\neg s_1 \wedge \neg s_2 \wedge ... \wedge \neg s_m) =$

$\prod_{i=\overline{1,m}}(1-P(s_i)) > (1-\gamma^n)^m > 1 - m\gamma^n$. Note that we used Bernoulli's inequality at the end. So, the probability of generating at least one copyrighted text is lower than $1-(1-m\cdot\gamma^n)$ and hence lower than $m\cdot\gamma^n$.

Example. Let's consider a "hard" UMD watermarking scheme with $\gamma = 0.5$. Let's assume each copyrighted text is 100 tokens, the model was trained on a dataset containing 10^9 copyrighted texts. The probability to generate a copyrighted text is $< 10^9 \cdot 0.5^{100} = \frac{10^9}{2^{100}} = \frac{1000^3}{1024^{10}} < \frac{1000^3}{1000^{10}} = 1000^{-7} = 10^{-21}$ and hence very low.

Theorem 2 *Let f be a LLM and f_W its watermarked version with a "soft" UMD scheme and let $\epsilon \in (0, \frac{1}{4})$. Let $s = t_1 \oplus t_2 \oplus ... \oplus t_n \in C_D$ be a copyrighted sample. We consider $\gamma = 0.5$. We denote the output of the softmax layer of f for generating the token t_i as $\frac{a_i}{d_i+a_i}$ and in the case of f_W, we denote it by $\frac{a_i \cdot e^\delta}{b_i^{'}+c_i^{'}\cdot e^\delta + a_i \cdot e^\delta}$ (if t_i is on the green list) and $\frac{a_i}{b_i^{''}+c_i^{''}\cdot e^\delta + a_i}$ (if t_i is on the red list), where a_i is the exponential of the logit value corresponding to the token t_i and $b_i^{'}, b_i^{''}$ and $c_i^{'}, c_i^{''}$ are the sum of the exponentials of the logits corresponding to other tokens that are on the red list and green list, respectively. We assume that $\frac{x}{a_i} < M = \frac{1-4\epsilon}{1+4\epsilon}$, for all $x \in \{d_i, b_i^{'}, b_i^{''}, c_i^{'}, c_i^{''}\}$ which would restrict f to be relatively confident in its predictions for each token t_i. Then, we can always find a δ (strength) for the watermarking scheme such that the probability of generating s is reduced by at least $(1+\frac{2\epsilon}{2\epsilon+1})^n$ times in comparison to the case of the unwatermarked model.*

Proof. First, we observe that the probability of generating the token t_i by the unwatermarked model is $\frac{a_i}{d_i+a_i} = \frac{1}{\frac{d_i}{a_i}+1} > \frac{1}{M+1} = 1/2 + 2\epsilon$.

We observe that since there is a finite number of $\frac{x}{a_i}'s$ and they are all positive, then it exist a lower bound for $\frac{x}{a_i}$ (let's denote it by $m > 0$). Since $\gamma = 0.5$, the probability of t_i being a green token is $\frac{1}{2}$ and hence the probability of the watermarked model to generate t_i is $\frac{1}{2}\frac{a_i \cdot e^\delta}{b_i^{'}+c_i^{'}\cdot e^\delta + a_i \cdot e^\delta} + \frac{1}{2}\frac{a_i}{b_i^{''}+c_i^{''}\cdot e^\delta + a_i} < \frac{1}{2} + \frac{1}{2}\frac{a_i}{b_i^{''}+c_i^{''}\cdot e^\delta + a_i} = \frac{1}{2} + \frac{1}{2}\frac{1}{\frac{b_i^{''}}{a_i}+\frac{c_i^{''}}{a_i}\cdot e^\delta + 1} \leq \frac{1}{2} + \frac{1}{2}\frac{1}{m\cdot(e^\delta+1)+1}$. We pick $\delta > log\big(\frac{1-2\epsilon(m+1)}{2\epsilon m}\big)$ and we observe that $\frac{1}{2} + \frac{1}{2}\frac{1}{m\cdot(e^\delta+1)+1} < \frac{1}{2} + \frac{1}{2}\frac{1}{m\cdot(\frac{1-2\epsilon(m+1)}{2\epsilon m}+1)+1} = \frac{1}{2} + \frac{1}{2}\frac{1}{m\cdot(\frac{1-2\epsilon}{2\epsilon m})+1} = \frac{1}{2} + \epsilon$.

So, by combining the two observations above, we conclude that the probability of generating t_i is reduced by at least $\frac{\frac{1}{2}+2\epsilon}{\frac{1}{2}+\epsilon} = 1 + \frac{2\epsilon}{2\epsilon+1}$ times. Therefore, since there are n tokens in s, the probability of generating s is reduced by at least $(1+\frac{2\epsilon}{2\epsilon+1})^n$ times.

Observation. Since the probability is reduced by at least $(1+\frac{2\epsilon}{2\epsilon+1})^n$ times in Theorem 2 then the probability of generating s is lower than $(\frac{2\epsilon+1}{4\epsilon+1})^n$ (as the maximum probability of generating with the unwatermarked model is 1). Hence, as in Theorem 1, we observe that the probability of generating a copyrighted text is lower than $m \cdot (\frac{2\epsilon+1}{4\epsilon+1})^n$.

Takeaways. Our theoretical analysis demonstrates that watermarking significantly reduces the probability of generating copyrighted text verbatim. For both a "hard" and "soft" UMD scheme, the upper bound for the likelihood of producing copyrighted content decreases exponentially with the length of the copyrighted texts.