Baselines for Identifying Watermarked Large Language Models

Leonard Tang [1] **Gavin Uberti** [1] **Tom Shlomi** [1]

Abstract

We consider the emerging problem of identifying the presence of watermarking schemes in publicly hosted, closed source large language models (LLMs). Rather than determine if a given text is generated by a watermarked language model, we seek to answer the question of if the model itself is watermarked. We introduce a suite of baseline algorithms for identifying watermarks in LLMs that rely on analyzing distributions of output tokens and logits generated by watermarked and unmarked LLMs. Notably, watermarked LLMs tend to produce token distributions that diverge qualitatively and identifiably from standard models. Furthermore, we investigate the identifiability of watermarks at varying strengths and consider the tradeoffs of each of our identification mechanisms with respect to watermarking scenario.

1. Introduction

Recent progress in large language models (LLMs) has improved their ability to produce convincingly human-like text. Models like GPT-4 (OpenAI, 2023b) and PaLM-2 (Anil et al., 2023) can perform at expert levels in many fields, sparking worries that LLMs could be used to spread disinformation. As such, distinguishing AI and human generated text has become a popular field of research. Yet current methods are shockingly fallible - OpenAI's detector has a false positive rate of 9% and a (self-reported) true positive rate of just 26% (Kirchner et al., 2023). Such poor performance makes these methods impractical for detecting student cheating or AI-generated spam emails.

One alternative is to *watermark* the text while it is being generated - subtly modifying it in a way that is indistinguishable to humans but detectable by algorithms. Several forms of watermarks have been introduced, subject to the requirement that the detection algorithm has sufficient access to a subset of the watermark's parameters (Kirchenbauer et al., 2023; Aaronson, 2023).

But we believe the deployment of watermark technology poses risks for consumer rights in several ways. First, watermark technology *must* degrade the quality of products themselves. As described in section two, watermarks like that proposed by (Kirchenbauer et al., 2023) randomly adjust a subset of the outputs, which inherently changes model outputs and thus quality. While the authors try to minimize quality degradation, any watermark strength above 0 must impact quality.

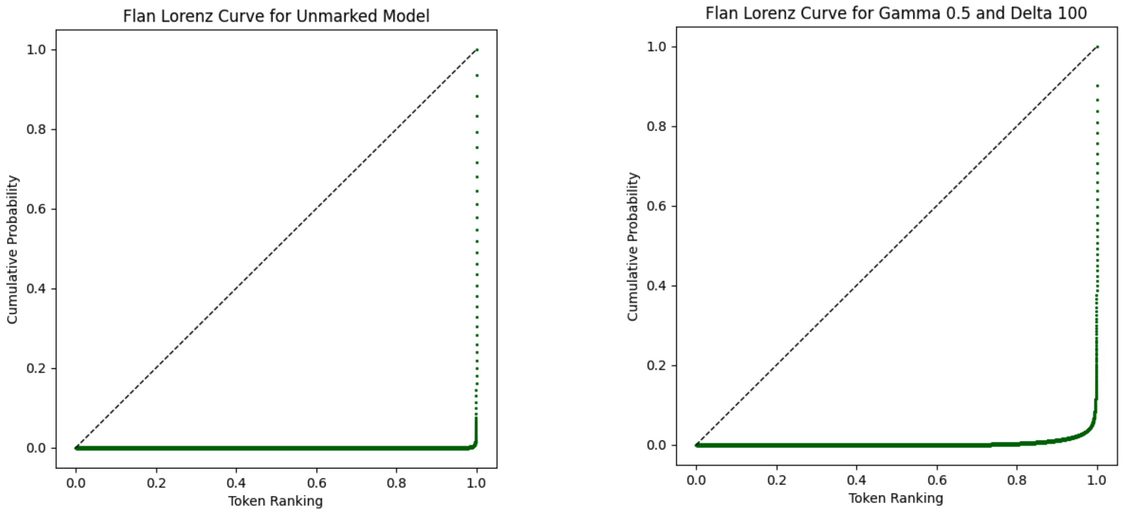
LLM watermarking also bears similarities to digital rights management (DRM) software in media, where the concern of unauthorized usage and distribution degraded the quality of users' products without their consent. While DRM software has many vocal opponents, with Apple famously removing DRM software from iTunes, there is almost no concern regarding the watermarking of LLMs. Motivated by these concerns, we are interested in identifying watermarked language models. Differing from previous work, where the focus is on determining if text has been produced by a watermarked model, here we study the problem of if a *language model* has been watermarked. Critically, our black-box algorithms only require querying the model and do not necessitate any knowledge of underlying watermarking parameters.

2. Related Work

Generated Text Detection Via Statistical Discrepancies Recent methods such as DetectGPT and GPTZero distinguish between machine-generated and human-written text by analyzing their statistical discrepancies (Tian, 2023; Mitchell et al., 2023). DetectGPT compares the log probability computed by a model on unperturbed text and perturbed variations, leveraging the observation that text sampled from a LLM generally occupy negative curvature regions of the model's log probability function. GPTZero instead uses perplexity and burstiness to distinguish human from machine text, with lower perplexity and burstiness indicating a greater likelihood of machine-generated text. However, these heuristics do not generalize and are often fallible.

[1] Department of Computer Science, Harvard University. Correspondence to: Leonard Tang <leonard-tang@college.harvard.edu>.

2^{nd} *AdvML Frontiers workshop at* 40^{th} *International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).



(a) Lorenz curve for a unmarked Flan-T5-XXL language model. Most of the probability mass is concentrated in a few top tokens, as visualized by the sharp spike towards the right of the Lorenz curve.

(b) Lorenz curve for a Flan-T5-XXL model affected by a Kirchenbauer watermark with parameters $\gamma = 0.5$ and $\delta = 100$. Notice that the Lorenz curve is slightly smoother under this setting, due to the δ application on low-probability tokens.

Figure 1. Examples of ranked probability Lorenz curves of the first token generated by Flan-T5-XXL under different Kirchenbauer watermarking strengths. The dashed line represents a perfectly uniform distribution. In both watermarking settings, the majority of the probability mass is concentrated in the top few tokens.

Detection by Learning Classifiers Several papers have proposed to train classifiers to distinguish between AI and human generated text. During the initial GPT-2 release, OpenAI trained a RoBERTa classifier to detect GPT-2 generated text with 95% accuracy (Solaiman et al., 2019). More recently, OpenAI fine-tuned a GPT model on a dataset of machine-generated and human texts focusing on the same topic, with a true positive identification rate of 26% (OpenAI, 2023a). Similarly, Guo et al. (2023) collected the Human ChatGPT Comparison Corpuse (HC3) and fine-tuned RoBERTa for the detection task.

Notably, the capabilities of such classifiers decrease as machine-generated text becomes increasingly human-like. Sadasivan et al. (2023) show theoretically that for sufficiently advanced language models, machine-generated text detectors offer only a marginal improvement over random classifiers. Moreover, such methods are prone to adversarial attacks and are not robust to out-of-distribution text.

Watermarking Large Language Models An alternative to detecting machine-generated text is watermarking. Watermarks are hidden patterns in machine-generated text that are imperceptible to humans, but algorithmically identifiable as synthetic. Natural language watermarks long predate the development of LLMs, relying on methods such as synonym substitution, as well as syntactic and semantic transformations (Topkara et al., 2005).

More recently, Kirchenbauer et al. (2023) proposed a watermarking scheme that minimizes degradation in the quality of generated text, while being efficient to detect in text. In ongoing work, Aaronson (2023) introduces a conceptually similar watermarking scheme. At any given inference step, both watermarking approaches modify the output token probabilities of the underlying model with an algorithm using a secret key, hashing, and pseudorandom function properties. We broadly refer to both of these watermarks as *Kirchenbauer watermarks*, which we develop identification mechanisms against.

Briefly, a Kirchenbauer watermark operates at decoding step t by first using the underlying LLM to generate a probability vector $p^{(t)}$ over a vocabulary V; next computing a hash from the *previous* token $s^{(t-1)}$ and seeding a random number generator (RNG); then using the RNG to partition V into two sub-vocabularies G (green list) and R (red list); followed by adding δ to all logits corresponding to G; and finally sampling token $s^{(t)}$ based on $p^{(t)}$ after the δ perturbation on G. For a green list proportion of $\gamma \in (0, 1)$ and a watermarked text of length T, this procedure expects a human, or otherwise LLM with no knowledge of the watermarking scheme, to use γT green list tokens with variance $T\gamma(1-\gamma)$. Defining the following null hypothesis H_0 : *The text sequence is generated with no knowledge of the water-*

Table 1. Tradeoffs between proposed watermarking identification algorithms. An ideal watermarking is not specific to Kirchenbauer and can detect **general watermarks**; does **not require access to logits**, which is common in publicly hosted models; is **sensitive to small** δ and parameter values; is **robust against other distribution shifts** not induced by watermarks; and can be **performed in a single snapshot of time** without reference to previous distributions or tests.

DETECTION METHOD	GENERAL WATERMARKS	LOGIT-FREE	δ-SENSITIVE	SHIFT-ROBUST	SINGLE-SHOT
RNG DIVERGENCE	√	√	√	×	×
MEAN ADJACENT	×	×	√	×	×
δ-AMPLIFICATION	×	×	√	√	√

mark's partitioning, one can then apply a one-proportion z-test to accept or reject H_0 with the following z-statistic:

$$z = (|s|_G - \gamma T)/\sqrt{T\gamma(1-\gamma)}$$

3. Baseline Mechanisms for Identifying Watermarked Large Language Models

Here, we introduce three baseline mechanisms for identifying the presence of watermarks in LLMs. These mechanisms are centered on analyzing the exact and approximate probability and logit distributions produced by a LLM. Critically, our mechanisms do not require any access to information governing the underlying watermark generation procedure, such as a hash function or random number generator.

Our three proposed algorithms vary in their access to exact versus sampled logits, generalizability across watermarking schemes, and statistical robustness. Depending on the objective and identification constraints, such as efficient computation, interpretable test statistic, availability of logits, and robustness to random shifts in the data, a different algorithm will be optimal. We hope these algorithms can serve as sound baselines for future work in this field.

3.1. Measuring Divergence of RNG Distributions

The first algorithm is centered on the simple idea of measuring divergence in "random" number distributions generated by a LLM. Specifically, we treat LLMs as random number generators, asking them to generate integers from 1 to 100:

```
"""Below is an instruction that
describes a task. Write a response that
appropriately completes the request.

### Instruction:
Generate a random number between
1 and 100.

### Response:"""
%
```

A key benefit of the random number generation task over alternatives is that the output space for any model is fairly consistent between models. While the distribution of numbers is certainly expected to change across models, the range of outputs is relatively more stable.

To detect if a specific LLM is watermarked, we first generate a 1000 number empirical distribution $F_{u,n}$ using a known *unmarked* model as described above. Then, we watermark the LLM and produce an empirical distribution $F_{w,m}$ in the same fashion. Finally, we compute the Kolmogorov-Smirnov statistic (Massey Jr, 1951) as follows to determine whether the empirical random number distribution of the LLM has shifted under watermark application:

$$D_{n,m} = \sup_x |F_{u,n}(x) - F_{w,m}(x)|$$

Here n and m are the sizes of each sample, and $n = m = 1000$ specifically in our case. The null hypothesis is that the samples are drawn from the same distribution, i.e.:

H_0: $F_{u,n}$ and $F_{w,m}$ are drawn from the same underlying distribution

We reject H_0 at significance level α if:

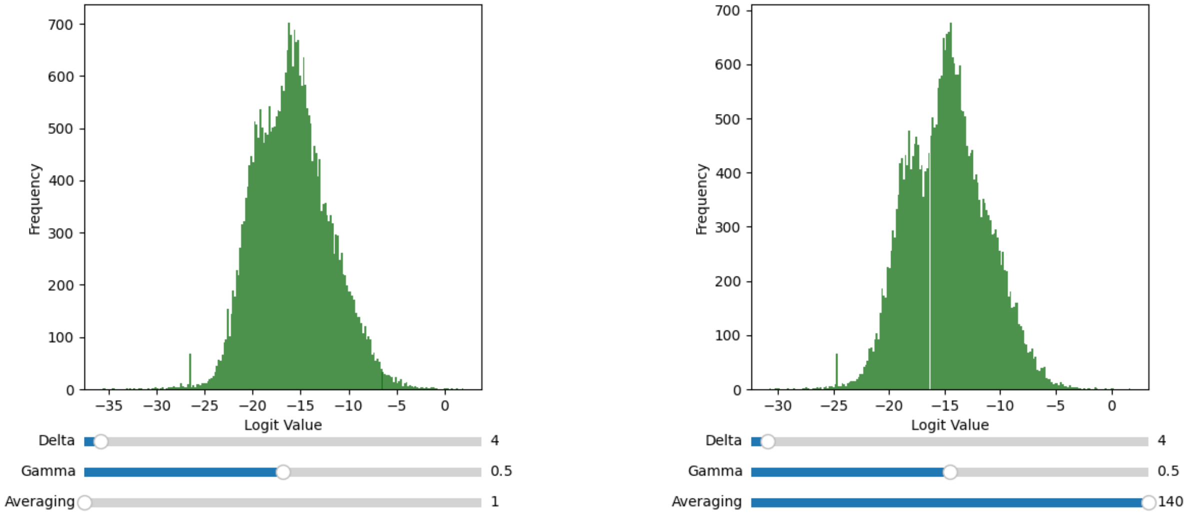
$$D_{n,m} > c(\alpha)\sqrt{\tfrac{n+m}{n \cdot m}}$$

Here $c(\alpha) = \sqrt{-\ln\left(\frac{\alpha}{2}\right) \cdot \frac{1}{2}}$, and we set the significance level $\alpha = 0.05$ in our experiments.

3.2. Mean Adjacent Token Differences

Inspired by econometrics, we use the Lorenz curve of model output probabilities to understand language model behavior. Specifically, we examine the output token probabilities of a model and construct *ranked probability Lorenz curves*. The x-axis of a ranked probability Lorenz curve lists the tokens sorted from lowest to highest probability, and the y-axis of the curve displays the probabilities of each token. Due to the sorted construction of the x-axis, the ranked token Lorenz curve is monotonically increasing. Figure 1 displays an example of these Lorenz curves.

The Lorenz curve is an effective tool for understanding



(a) Distribution of logit values prior to δ-Amplification.

(b) Distribution of logit values after applying δ-Amplification, averaging across 140 prompts.

Figure 2. Distribution of Alpaca-LoRA logit values before and after δ-Amplification for the "Now write me a story:" prompt at a small δ value of 4. The x-axis is the logit value and y-axis is the frequency. Without δ-Amplification, it is not clear whether the distribution of logit values exhibits bimodality. Bimodality emerges only after applying δ-Amplification, enabling watermark identification.

the effects of a Kirchenbauer watermark. In the ranked token Lorenz curve, the watermark produces manifests a smoothing effect, as seen on the right of Figure 1, indicating that a portion of lower-probability tokens have experienced a δ-increase.

To rigorize this notion of smoothness, one can compute the Gini coefficient G of the Lorenz curve:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \overline{x}}$$

Here x_i, x_j are the probabilities of i-th and j-th tokens on the curve, indexed by the ordered ranking, and $\overline{x}$ is the average probability. Traditionally in economics, G is used to measure the inequality of a distribution. High G suggests more inequality, reflected in unmarked distributions, while low G suggests less inequality and a smoother distribution, suggesting the presence of a watermark.

A natural extension is to analyze the average increase in logit value between adjacent tokens. That is, we compute:

$$\mathcal{I} = \frac{\sum_{i=1}^{n-1} \ell_{i+1} - \ell i}{n-1}$$

Here, ℓ_i is the logit at index i on the Lorenz curve, and n is the total number of tokens in the vocabulary.

Note that for a Kirchenbauer-watermarked LLM with logit perturbation δ and green list G with proportion γ, we have an average logit increase of:

$$\begin{aligned}\mathcal{I}_{\mathcal{W}} &= \frac{\sum_{i=1}^{n-1} (\ell_{i+1} - \ell i) \mathbb{1}[i \in G]}{n-1} \\ &= \frac{\gamma(n-1)\delta + \sum_{i=1}^{n-1} (\ell_{i+1} - \ell i)}{n-1}\end{aligned}$$

Taking the difference with the average logit increase of an unmarked model, $\mathcal{I}_{\mathcal{U}}$, we have:

$$\mathcal{I}_{\mathcal{W}} - \mathcal{I}_{\mathcal{U}} = \frac{\gamma(n-1)\delta}{n-1} = \gamma\delta$$

Taking the above $I_W - I_U$ as inspiration, a simple identification procedure is to periodically compute $\mathcal{I}$ and observe how it varies over time. Notice that $I_W - I_U$ directly varies with γ and δ; that is, the strength of the watermark directly influences its detectability. For a strong watermark, variations in $\mathcal{I}$ will be obvious, while weaker watermarks will manifest subtler differences in $\mathcal{I}$.

3.3. Robustly Identifying Small-δ Watermarks

While §3.2 introduces a metric that will successfully detect a Kirchenbauer watermark for small δ, it is sensitive to general logit distribution perturbations introduced by other scenarios, such as routine model updates. An identification method robust to general distribution shifts should rely on shift characteristics specific to a Kirchenbauer watermark.

Table 2. δ-Amplification algorithm produces the corresponding bimodality test dip and p-values for a small-δ watermarked Alpaca-LoRA model when using prompt prefixes randomly sampled from the Pile and OpenWebText datasets. Greater diversity in prompt prefix task and content increasingly induces bimodality and thus watermark identification. Notice that when only using Pile prompts, δ-Amplification is only identify a watermarked model at strength $\delta = 7$, compared to $\delta = 5$ when using both OWT and Pile prompts.

δ	P-VALUE (OWT & PILE)	DIP (OWT & PILE)	p (PILE)	DIP (PILE)
0	0.886	0.0017	0.908	0.0016
1	1.0	0.00094	0.999	0.00097
2	0.991	0.0013	1.0	0.00092
3	0.900	0.0016	1.0	0.00093
4	0.204	0.0025	1.0	0.00093
5	0.0	0.00497	1.0	0.00093
6	0.0	0.0087	0.947	0.0015
7	0.0	0.012	0.0	0.0048
8	0.0	0.017	0.0	0.013
9	0.0	0.023	0.0	0.025
10	0.0	0.033	0.0	0.037

A consequence of the Kirchenbauer watermark is that it induces perceptible separations of logit mass, where the separation magnitude is of size δ. Inspired by this observation, we draw an analogue between the separation of logit values into bands and the bimodality of logit frequencies. Under this reframing, testing for bimodality, such as in Figure 2, is equivalent to testing for the existence of a band gap.

However, though this approach is robust to other distribution shifts, it does not yet consider small-δ perturbations. To handle such situations, we introduce the δ-Amplification algorithm.

Algorithm 3.1 (δ-Amplification)**.** *Suppose we have a potentially watermarked LLM $\mathcal{L}$. We wish to detect if it is watermarked. We prompt $\mathcal{L}$ repeatedly as follows:*

```
[Random string sampled from training
datasets]. Now write me a story:
```

Take the produced logits and average them across repetitions. If the resulting frequency of averaged logits is bimodal, conclude that $W_s(\mathcal{L})$ is watermarked.

To recover the underlying Kirchenbauer watermark parameters, we estimate δ by measuring the distance between the peaks, and γ by measuring their respective masses.

Critically, as watermarks (Kirchenbauer et al., 2023; Aaronson, 2023) only use a fixed-size previous token window (rumored to be 5-tokens in OpenAI models) to determine green list indices, the green list partition across all prompts is the same under this algorithm, as every prompt ends in a fixed “Now write me a story:” suffix. Therefore, the output logit distributions all experience the same δ mask.

However, the model is still influenced by earlier tokens in the prompt, and thus exhibits differing logit values across prompts. Intuitively then, averaging distributions across different prompts reduces the variation of logits, but maintains the same effect of the δ perturbation. The averaged distribution thus amplifies the effects of a small-δ watermark. Figure 2 demonstrates an example of this effect.

We test for bimodality via the Hartigan dip test (Hartigan & Hartigan, 1985). For a distribution with probability distribution function f, this test computes the largest absolute difference between f and the unimodal distribution which best approximates it.

$$D(f) = \inf_{g \in U} \sup_{x} |f(x) - g(x)|$$

Here U is the set of all unimodal distributions over x. The corresponding p-value is calculated as the probability of achieving a Dip score at least as high as D from the nearest unimodal distribution.

3.4. Tradeoffs Between Detection Algorithms

The algorithms proposed above are all effective in different senses. §3.1 introduced a RNG divergence approach to watermark detection that is not specific to Kirchenbauer watermarks, does not require access to logits and can thus be used directly on black-box public APIs, and is also sensitive to small δ watermarks.

§3.2 introduced an adjacent token metric for analyzing Kirchenbauer watermarks that is sensitive to small-δ.

Finally, we extended this approach in §3.3 to robustly handle small-δ watermarks, while preserving the single-shot criteria. Moreover, the δ-Amplification approach lent itself nicely to statistical testing, specifically of bimodality. It is also the only identification method that can be performed in a single shot. That is, it does not require comparing behavior between a watermarked and unmarked model, and equivalently between a single model across multiple snapshots in time, as is the case for the previous two algorithms.

Each method has merit depending on the specific identification setting, but will also sacrifice certain desiderata. Table 1 summarizes these tradeoffs.

4. Results

We perform experiments on our identification mechanisms using the Flan-T5-XXL and Alpaca-LoRA models due to their strong instruction-following capabilities but differing Byte-Pair Encoding and digit tokenization methods.

Table 3 displays the p-values resulting from the Kolmogorov-Test method. For each model and watermark strength, the method is performed across 30 independent instances of 1000-sample distributions generated from a Kirchenbauer-watermarked model. Specifically, we perform a test between each of the 30 distributions and a distribution generated by an unmarked model. Under this procedure, any model with distributions producing an average p-value less than $\alpha = 0.05$ would be considered watermarked. The p-values are highest when comparing an unmarked distribution against an unmarked distribution, as expected. Notably, the p-values are extremely low for a majority of watermark strengths for both Flan-T5-XXL and Alpaca-LoRA.

The results of the δ-Amplification method and corresponding bimodality test are in Table 2. Concretely, we sample a diverse range of prompt prefixes from Pile and OpenWebText via HuggingFace datasets and run tests on the logit value distributions from these generations. Notably, diversity in prompt prefix task and content enables uncorrelated variance in δ, thus most effectively eliminating logit variance post-averaging.

We observe that at $\delta \geq 5$, our method produces p-values less than $\alpha = 0.05$, thus successfully identifying the presence of a watermark. Critically, increasing the number of varied prefix prompts also increases identification potency - averaging logit distributions only across Pile prompts identifies Kirchenbauer-watermarked models only at strength $\delta \geq 7$, while averaging across both Pile and OpenWebText prompts identifies watermarked models at strength $\delta \geq 5$.

Table 3. Kolmogorov-Smirnov test results on 1000-sample “RNG“ distributions from models watermarked at varying strengths. A test is performed between 30 distributions generated from the model against a random distribution generated from an unmarked model. The reported p-values are averaged across these 30 samples. Rows where $\gamma = \delta = 0$ compare an unmarked model against an unmarked model.

MODEL	γ	δ	AVERAGE P-VALUE
FLAN-T5-XXL	0	0	0.80
FLAN-T5-XXL	0.1	1	3.54E-7
FLAN-T5-XXL	0.1	10	1.22E-9
FLAN-T5-XXL	0.1	50	6.75E-7
FLAN-T5-XXL	0.1	100	8.11E-8
FLAN-T5-XXL	0.25	1	0.002
FLAN-T5-XXL	0.25	10	6.47E-9
FLAN-T5-XXL	0.25	50	2.59E-7
FLAN-T5-XXL	0.25	100	1.33E-6
FLAN-T5-XXL	0.5	1	0.00024
FLAN-T5-XXL	0.5	10	0.057
FLAN-T5-XXL	0.5	50	0.054
FLAN-T5-XXL	0.5	100	0.054
FLAN-T5-XXL	0.75	1	0.42
FLAN-T5-XXL	0.75	10	0.22
FLAN-T5-XXL	0.75	50	0.34
FLAN-T5-XXL	0.75	100	0.23
ALPACA-LORA	0	0	0.63
ALPACA-LORA	0.1	1	1.40E-10
ALPACA-LORA	0.1	10	4.31E-37
ALPACA-LORA	0.1	50	4.31E-36
ALPACA-LORA	0.1	100	1.48E-35
ALPACA-LORA	0.25	1	3.10E-13
ALPACA-LORA	0.25	10	1.93E-10
ALPACA-LORA	0.25	50	4.52E-14
ALPACA-LORA	0.25	100	2.14E-11
ALPACA-LORA	0.5	1	4.17E-13
ALPACA-LORA	0.5	10	0.0089
ALPACA-LORA	0.5	50	0.066
ALPACA-LORA	0.5	100	0.06
ALPACA-LORA	0.75	1	0.00015
ALPACA-LORA	0.75	10	0.52
ALPACA-LORA	0.75	50	0.40
ALPACA-LORA	0.75	100	0.48

5. Conclusion

In this work, we develop a framework for understanding the novel problem of identifying watermarked large language models. Rather than detect if text is generated by a watermarked LLM, we detect if the LLM itself is watermarked. To that end, we provide three black-box baseline mechanisms – measuring divergence of RNG distributions, mean adjacent token differences in logits, and δ-Amplification – all of which fundamentally rely on the analysis of the distributions of model outputs, logits, and probabilities.

All of our techniques rely on knowledge about the behavior of the underlying models, which can take many forms - knowledge that the distribution of logits in transformer models is unimodal or knowledge that LLMs sample integers at certain probabilities. Schemes to remove bias from LLMs could make them more vulnerable to these attacks - if there was a query that caused a LLM to return “male” and “female” with equal probability (e.g. “Guess the gender of a barrister”), we could apply a similar scheme as used in section 3.1. We hope our work serves as a baseline for future watermark detection techniques.

References

Aaronson, S. My AI safety projects at OpenAI. Talk given to the Harvard AI Safety Team, March 2023.

Anil, R., Dai, A. M., Firat, O., et al. Palm 2 technical report, 2023.

Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., and Wu, Y. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*, 2023.

Hartigan, J. A. and Hartigan, P. M. The dip test of unimodality. *The annals of Statistics*, pp. 70–84, 1985.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models. *arXiv preprint arXiv:2301.10226*, 2023.

Kirchner, J. H., Ahmad, L., Aaronson, S., and Leike, J. New ai classifier for indicating ai-written text, Jan 2023. URL https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Massey Jr, F. J. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46 (253):68–78, 1951.

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., and Finn, C. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*, 2023.

OpenAI. New AI classifier for indicating AI-written text. https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text, 2023a.

OpenAI. Gpt-4 technical report, 2023b.

Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., and Feizi, S. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.

Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Kim, J. W., Kreps, S., et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

Tian, E. GPTZero. https://gptzero.me/, 2023.

Topkara, M., Taskiran, C. M., and Delp III, E. J. Natural language watermarking. In *Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pp. 441–452. SPIE, 2005.