LLM WATERMARKING USING MIXTURES AND STATISTICAL-TO-COMPUTATIONAL GAPS

PEDRO ABDALLA AND ROMAN VERSHYNIN

ABSTRACT. Given a text, can we determine whether it was generated by a large language model (LLM) or by a human? A widely studied approach to this problem is watermarking. We propose an undetectable and elementary watermarking scheme in the closed setting. Also, in the harder open setting, where the adversary has access to most of the model, we propose an unremovable watermarking scheme.

1. INTRODUCTION

Large Language Models (LLMs) have emerged as a powerful technology for generating human-like text [3, 18]. On one side, an LLM performs well if it produces text that closely resembles human writing. On the other side, malicious use of high-performance LLMs also bring undesirable consequences such as the spread of misinformation [17], misuse in education [13, 18], and data pollution [15, 16].

In this context, there is an urge to develop methods to distinguish human and AI generated text to mitigate those outcomes. One prominent technique is the so-called watermarking approach in which the goal is to embed a detectable signal in the text generated by the LLM.

Before describing watermarking in more details, we recall the concept of tokenization. In a nutshell, a word consists in small pieces of "sub-words" known as tokens. A LLM outputs each token sequentially by computing a probability distribution over a fixed set of possible tokens (vocabulary) and sampling the next token from it. The distribution of the next token varies from token to token as it depends on the previous tokens sampled, while the vocabulary remains fixed during the process of text generation.

The most common approach to watermark a text is to watermark each token by planting a hidden structure into its probability distribution. In this sense, it is natural to impose some requirements on which properties a good watermarking should have. For example, it is natural to require that the watermarking scheme does not deteriorate the quality of the text or that it cannot be easily removed by someone with malicious intentions, an adversary.

In what follows, we describe the requirements for our watermarking scheme. To this end, we shall make a distinction between two different settings: The closed setting and the open source setting. We first describe the closed setting. In this case, one would like to generate a watermarking satisfying three requirements

- **Undetectability:** Any polynomial-time algorithm based solely on the text generated by the LLM fails to detect any change in the probability distributions used to generate the tokens.

Department of Mathematics, UC Irvine.

- **Completeness:** It is possible to detect the watermarked model if the algorithm has access to extra piece of information known as "secret key".
- **Soundness:** Any text generated independently of the secret key has negligible chance of being detected as watermarked.

We postpone the mathematical framework of those requirements to the next section. Now, let us provide some intuition behind those requirements. The first requirement is useful to preserve the quality of the text generated and prevents malicious users (adversaries) to manipulate the text to remove the watermarking scheme. The second requirement is the core idea of watermark to distinguish texts generated by AI and humans which clearly requires a "secret key", otherwise would contradict the "undetectability" requirement. Finally, the last requirement is of fundamental importance as, for example, it prevents false accusations of AI misuse (see for example [7]).

A harder task is to watermark the text in the so-called open source setting. This is motivated by the recent explosion of AI open source models, where the user has access to the model parameters and the associated code [5, 19, 23]. Since now the adversary has much more power, we replace the "undetectability" requirement by the weaker "unremovability" requirement:

- **Unremovability:** Any adversary that does not have knowledge about the secret key cannot remove the watermarking scheme unless it deteriorates the quality of the text.

Clearly, one has to impose some conditions on what the adversary knows, otherwise he could train a new model on its own, making watermarking impossible. Besides, the adversary goal is to remove the watermark and use the text for malicious purposes, so the quality of the text cannot be deteriorated too much.

In this work, we allow the adversary to arbitrarily modify the inputs of the LLM and also allow him to have knowledge of each token distribution used for sampling (after the watermarking scheme was planted).

1.1. **Related Work.** Several watermarking schemes were proposed [2, 14, 22] for the closed setting without any formal guarantee. Perhaps, the first watermarking scheme with provable guarantees is from [9], where the authors proposed to split the vocabulary into a green list and red list. The probabilities corresponding to the tokens in the green list are slightly increased while the ones in the red list are slightly decrease. The watermarking can be detected by checking the frequency of tokens in the green list versus tokens in the red list. Therefore, the downside of this approach is that the "undetectability" requirement is not fulfilled.

Another line of work [1,6,10,12] is dedicated to the following idea: Let u be a random variable distributed uniformly over the interval $[0, 1]$ and $p = (p(1), \ldots, p(d))$ be a probability distribution over a vocabulary of size d. We can sample the next token according to p by sampling from u first and then observing that for any $k \in \{1, \ldots, d\}$

$$\mathbb{P}\left\{u \in \left[\sum_{i=1}^{k-1} p(i), \sum_{i=1}^{k} p(i)\right]\right\} = p(k).$$

The watermarking schemes exploit correlation between the tokens and the corresponding (uniform) random variables u's used to sample the tokens. Despite this approach has some guarantees, the major drawback is that the detection algorithm is quite convoluted relying on a complicated optimization because it is hard to

capture the planted correlation. In addition to this, to achieve the "undetectability" requirement for the whole text, the approaches in the literature rely on some cryptographic assumptions.

To the best of our knowledge, the only result for the open source setting is from [5]. The authors proposed to perturb each logit in the softmax rule (see equation 2.1) by a vector sampled from a multivariate Gaussian distribution and exploits the correlation between the text and such vector. The authors provided some theoretical evidence (partially rigorous) for completeness and unremovability under strong assumptions on the text.[1]

1.2. **Main Contributions.** Our main contributions are new connections between watermarking and robust statistics to derive more efficient watermarking schemes. Our first main result is Theorem 3.2, where we propose an elementary watermarking scheme for the closed setting satisfying all the requirements (undetectability, completeness and soundness) under mild assumptions on the distribution of the text. We also argue that the assumptions are necessary in some sense.

In a nutshell, our watermarking scheme proceeds as follows: In the first step it randomly constructs partitions of the vocabulary set into green and red tokens that change at each time a new token is sampled. Similarly to [9], the probabilities of the green tokens are shifted upwards, and the ones for the red tokens, downwards. However, we make some key changes to this shifting scheme, which allows it to achieve undetectability.

Our second main result Theorem 4.3 lies in the realm of the open source setting. By leveraging novel connections to the theory of statistical-to-computational gaps in robust statistics, we proposed a watermarking scheme that is both "sound" and "complete", along with mathematically rigorous guarantees. This watermark is also "unremovable": any algorithm that attempts to remove it must (indirectly) solve a computationally hard problem – the sparse mean estimation under Huber's contamination model (see [8] for a comprehensive introduction).

This version of our algorithm also perturbs the logits by Gaussian vectors, similarly to [5]. However, instead of using the same perturbation for each token, we use independent perturbations.

Our Gaussian random perturbations are drawn from a mixture of non-centered Gaussians – a distribution that is hard to distinguish from a centered Gaussian. Thus, the adversary who attempts to remove it faces impossibility results from the robust statistics literature borrowed from Brennan and Bresler [4]. This is a novel approach to study unremovability in watermarking problems.

1.3. **Roadmap.** The rest of this manuscript is organized as follows. In Section 2, we formally state the watermarking problem in the framework of hypothesis testing. Section 3 is dedicated to the main results for watermarking in the closed setting and Section 4 is dedicated to the main result for watermarking in the open source setting. The appendix is dedicated to the proof of technical results.

---

[1]For example, [5] assumes that the token distribution behaves as an uniform distribution over a certain subset and also that the adversary makes changes respecting some normalization of the soft-max function which are hard to verify.

2. The Hypothesis Testing Formulation

Let T be our vocabulary of d tokens, which we can identify with $[d] \coloneqq \{1, \ldots, d\}$ without loss of generality. A text is a sequence of random variables x_i taking values in $[d]$. At each step, the LLM computes logits $L = (\ell(1), \ldots, \ell(d))$ and samples the next token $x \in [d]$ according to the softmax rule

$$(2.1) \qquad p(i) \coloneqq \mathbb{P}\{x = i\} = \mathrm{softmax}(\ell(1), \ldots, \ell(d)) \coloneqq \frac{e^{\ell(i)}}{e^{\ell(1)} + \cdots + e^{\ell(d)}}.$$

A watermarking scheme consists of two parts:

- The *sampling algorithm* at each step tweaks the LLM's output probabilities from $p = (p(1), \ldots, p(d))$ to a new (watermarked) distribution $q = (q(1), \ldots, q(d))$, using a "secret key".
- The *detection algorithm* takes the whole text $x_1, \ldots, x_N$ and the same secret key, and outputs true/false depending on whether the text was watermarked.

The sampling algorithm handles undetectability by making sure that q looks like p. The detection algorithm handles soundness and completeness by solving the following hypothesis testing problem:

Detection Algorithm Hypothesis Testing:

- H_0 : The text $x_1, \ldots, x_N$ is independent from the watermarking scheme.
- H_a : The distribution of the text $x_1, \ldots, x_N$ was sampled from the watermarked distribution.

The core challenge in the design of a watermarking scheme is that we need to balance the trade-off between undetectability and completeness without using any prior knowledge about the text. If we simply do not change the LLM's distribution, then the scheme is undetectable but not complete. Similarly, if we make some obvious change to the LLM's distribution, the scheme is complete but detectable.

Remark 2.1 (Closed vs. Open Source Setting). We just covered the closed watermarking setting. In the open setting, the definition of watermarking as well as the "completeness" and "soundness" requirements stay the same. But the adversary now has more information than just the text. So instead of "undetectability", we require "unremovability".

We allow the adversary to modify anything they want about the LLM – the input, the parameters, even its its architecture. They can also modify the distributions $q_{x_1}, \ldots, q_{x_N}$ used to generate the watermarked text $x_1, \ldots, x_N$ – but the adversary is not given access to the secret key.

The goal of the adversary is to approximate the distributions $p_{x_1}, \ldots, p_{x_N}$. The "unremovability" requirement prevents the adversary from achieving this in polynomial time.

3. Watermarking Scheme – Closed Setting

Now, let's formally describe the watermarking scheme in the closed setting. It works for any discrete process on a finite state space, but we will stick to the LLM setting to keep things concrete.

Let N be the length of the text to be generated[2] and d be the size of the token vocabulary. Without loss of generality we assume that d is even.[3] Next, let r be a Rademacher random variable (± 1 with equal probabilities) and consider the alternating signs vector $\Delta := (r, -r, r, -r, \ldots, r)$ of length d. The secret key consists on i.i.d. copies $\Delta_1, \ldots, \Delta_N$ of the random vector Δ, one for each token in the text $x_1, \ldots, x_N$ to be watermarked.

For simplicity of notation, we drop the index j while describing the j-th step of the sampling algorithm. At each new step, the LLM computes the probability distribution of the next token x, namely $p = (p(1), \ldots, p(d))$. Now the sampling algorithm watermarks this distribution by increasing the probabilities $p(i)$ for the tokens i where $\Delta(i) > 0$ and decreasing them where $\Delta(i) < 0$.

To do that, we first assume that $p(1) \geq \cdots \geq p(d)$. Define for each $i \in [d]$:

$$(3.1) \qquad \varepsilon_i = \begin{cases} p(i), & i \text{ is even} \\ p(i+1), & i \text{ is odd}, \end{cases}$$

so that these numbers are pairwise equal: $\varepsilon_1 = \varepsilon_2$, $\varepsilon_3 = \varepsilon_4$, $\ldots$, $\varepsilon_{d-1} = \varepsilon_d$. Next, we compute the watermarked distribution by setting for every $i \in [d]$:

$$(3.2) \qquad q(i) = p(i) + \varepsilon_i \Delta(i).$$

In the general case, we first compute a non-increasing rearrangement of p, perturb it as before, and rearrange back.

Now let's check that this watermarking scheme is valid (i.e. q is actually a probability distribution) and undetectable.

Proposition 3.1 (Validity and undetectability). *For any probability distribution p over $[d]$, the watermarked distribution q computed in (3.2) is indeed a probability distribution over $[d]$, and the watermarking satisfies the undetectability requirement.*

Proof. Sine Δ alternates signs and $\varepsilon_1, \ldots, \varepsilon_d$ are pairwise equal, we have

$$\sum_{i=1}^{d} \varepsilon_i \Delta(i) = \Delta(1) \Big(\underbrace{(\varepsilon_1 - \varepsilon_2)}_{=0} + \underbrace{(\varepsilon_3 - \varepsilon_4)}_{=0} + \ldots + \underbrace{(\varepsilon_{d-1} - \varepsilon_d)}_{=0} \Big) = 0.$$

It follows that

$$\sum_{i=1}^{d} q(i) = \sum_{i=1}^{d} p(i) + \sum_{i=1}^{d} \varepsilon_i \Delta(i) = 1.$$

So, to check that q defines a probability distribution, it remains to show that it has nonnegative entries. But this follows from our construction: $q(i) \geq p(i) - p(i) = 0$ for even i, and $q(i) \geq p(i) - p(i+1) \geq 0$ for odd i.

Now let's check undetectability. For every $i \in [d]$, the probability that $x = i$ under q is

$$\mathbb{P}_q\{x = i\} = (p(i) + \varepsilon_i)\mathbb{P}\{\Delta(i) = 1\} + (p(i) - \varepsilon_i)\mathbb{P}\{\Delta(i) = 0\}$$
$$= \frac{1}{2}(p(i) + \varepsilon_i) + \frac{1}{2}(p(i) - \varepsilon_i) = p(i).$$

---

[2] Of course, the exact value of N is unknown but we can always work with the maximum number of tokens allowed by the LLM.

[3] In the case that d is odd, we simply add a spurious token

Because at each step $j = 1, \ldots, N$ we sample an independent copy of Δ, the distribution of the text $x_1, \ldots, x_N$ sampled from the watermarked distributions q and the unwatermarked distributions p remains the same. $\qquad\square$

3.1. **Watermarking Detection − Closed setting.** We now describe how our detection algorithm works. The core idea is that whenever the perturbation $\Delta(x)$ is positive, it increases the chance of the token x to appear in the text. Thus, it is more likely to observe the tokens in the text that correspond to the positive entries of Δ.

So, our algorithm just counts the fraction of tokens x_j in the text that have positive $\Delta_j(x_j)$, and tests if it significantly exceeds $1/2$:

Algorithm 1 Watermark Detection

Input: The text $x_1, \ldots, x_N$. The secret key: $\Delta_1, \ldots, \Delta_N$. Tolerance δ.
Output: True or False.

$Z \leftarrow \frac{1}{N} \sum_{j=1}^{N} \mathbb{1}_{\{\Delta_j(x_j)=1\}}.$

if $Z \geq 1/2(1 + \sqrt{3\log(1/\delta)/N})$ **then**
    return: True
end if
    return: False

Now, we state that the watermarking we just described is undetectable, sound, and complete:

Theorem 3.2 (Watermarking in Closed Setting)**.** *The watermarking scheme described above satisfies the following properties:*

(1) It is undetectable.

Moreover, for any $\delta \in (0,1)$, if $N \geq 3\log(1/\delta)$, then:

(2) It is sound with probability at least $1 - \delta$.
(3) Let p_j^* denote the probability of the most likely token in the vocabulary at step $j = 1, \ldots, N$. If, for some $\gamma > 1$,

$$(3.3) \qquad 1 + \frac{1}{N}\sum_{i=1}^{N}(1 - p_j^*) \geq \gamma\left(1 + \sqrt{\frac{3\log(1/\delta)}{N}}\right),$$

then the watermarking scheme is complete with probability at least $1 - e^{-c_\gamma N}$, where $c_\gamma := (1 - \gamma)/8$.

The result follows by taking $t = \gamma/2$ and using (4.3).

Remark 3.3 (Minimal entropy allows watermarking)**.** The assumption (3.3) on the distributions of the tokens is nearly necessary. If p_j^* gets too close to 1, it means that the token is almost deterministic, so watermarking it is impossible. The assumption (3.3) ensures that at least some fraction of the text is non-deterministic, making watermarking possible in principle.

Proof. We already proved *undetectability* in Proposition 3.1.

Soundness: Suppose we are under the null hypothesis H_0 that the text $x_1, \ldots, x_N$ is independent of the secret key $\Delta_1, \ldots, \Delta_N$. Then the test Z defined in Algorithm

1 has binomial distribution $(N, 1/2)$. By Chernoff's inequality, for every $t > 0$ we have

$$\mathbb{P}\{Z \geq 1/2(1+t)\} \leq e^{-t^2 N/3},$$

which is at most δ for $t = \sqrt{3\log(1/\delta)/N}$.

Completeness: Assume that the text is watermarked. Thus, at each fixed step, a random token x is picked from the distribution q defined in (3.2). Without loss of generality, assume that $p(1) \geq p(2) \geq \cdots \geq p(d)$. Then,

$$\mathbb{P}\{\Delta(x) = 1\} = \mathbb{E}\mathbb{1}_{\{\Delta(x)=1\}} = \mathbb{E}\sum_{i=1}^{d} \mathbb{1}_{\{\Delta(i)=1\}}\mathbb{1}_{\{x=i\}}$$

$$= \mathbb{E}\left[\sum_{i=1}^{d} \mathbb{1}_{\{\Delta(i)=1\}}\,\mathbb{E}\big[\mathbb{1}_{\{x=i\}}\,|\,\Delta\big]\right]$$

$$= \mathbb{E}\left[\sum_{i=1}^{d} \mathbb{1}_{\{\Delta(i)=1\}}\,\mathbb{P}\{x = i\,|\,\Delta\}\right]$$

$$= \mathbb{E}\sum_{i=1}^{d} \mathbb{1}_{\{\Delta(i)=1\}}\big(p(i) + \varepsilon(i)\big)$$

(by (3.2), since only the terms with $\Delta(i) = 1$ contribute to the sum)

$$= \sum_{i=1}^{d} \mathbb{P}\{\Delta(i) = 1\}\big(p(i) + \varepsilon(i)\big) = \frac{1}{2}\sum_{i=1}^{d}\big(p(i) + \varepsilon(i)\big)$$

(since each $\Delta(i)$ has Rademacher distribution)

$$= \frac{1}{2}\sum_{i=1}^{d} p(i) + \frac{1}{2}\sum_{i=1}^{d}\varepsilon(i)$$

$$= \frac{1}{2} + \frac{1}{2}\Big(p(2) + \underbrace{p(2)}_{\geq p(3)} + p(4) + \underbrace{p(4)}_{\geq p(5)} + \cdots + p(d)\Big)$$

(by construction (3.1) and the monotonicity assumption)

$$\geq \frac{1}{2} + \frac{1}{2}(1 - p(1)).$$

Next, set $Z_0 := 0$ and notice that

$$Z_k := Z_{k-1} + \mathbb{1}_{\{\Delta_k(x_k)=1\}} - \mathbb{E}\mathbb{1}_{\{\Delta_k(x_k)=1|x_1,\ldots,x_{k-1}\}},$$

is a martingale satisfying that $|Z_{(k)} - Z_{(k-1)}| \leq 1$ almost surely. Thus, by combining the Azuma-Hoeffding's inequality

$$\mathbb{P}\left\{Z \geq \frac{1}{2}\left(1 + \frac{1}{N}\sum_{i=1}^{N}(1 - p_j^*)\right) - t\right\} \geq 1 - e^{-t^2 N/2},$$

with the assumption (3.3), the result follows for $t = (\gamma - 1)/2$. $\qquad\square$

4. Watermarking Scheme - Open Source Setting

Unfortunately, in the open source setting, there is no guarantee that an adversary cannot destroy our watermarking scheme. So we propose a completely different approach in this setting.

We randomly perturb the logits in (2.1), similarly to [5]. However, the perturbation vector in [5] was sampled from a multivariate Gaussian distribution (the same at each step), while we propose to draw perturbation vectors from a random Gaussian mixture, which changes independently at each step.

Let's describe the construction of the perturbation vector $\Delta \in \mathbb{R}^d$ in detail. First, pick a k-sparse subset $S \subset [d]$ uniformly at random, and compute the vector

$$\mu = k^{-1/2} \mathbb{1}_S$$

supported on S. Let r be a Rademacher random variable and $G \sim N(0, I)$. For a fixed $\varepsilon > 0$, define the perturbation vector

$$(4.1) \qquad \Delta := \begin{cases} \varepsilon(G + \mu), & \text{if } r = 1 \\ \varepsilon(G - \mu), & \text{if } r = -1. \end{cases}$$

The role of the tuning parameter ε is to control the tradeoff between detectability of the watermark and the quality of the text. The *secret key* is the collection of vectors $G_1, \ldots, G_N$ used to generate the i.i.d. copies $\Delta_1, \ldots, \Delta_N$ of Δ at each step according to (4.1).

At each new step, the LLM computes the logits $\ell(1), \ldots, \ell(d)$, and our watermarking algorithm samples the new token according to the watermarked softmax rule given by

$$(4.2) \qquad q(i) := \frac{e^{\ell(i) + \Delta(i)}}{e^{\ell(1) + \Delta(1)} + \cdots + e^{\ell(d) + \Delta(d)}},$$

which is just a perturbed version of (2.1). We allow the adversary to have knowledge of the distribution q.

Intuitively, in order to remove the watermarking scheme, the adversary needs to guess the values of the perturbation vectors $\Delta_1, \ldots, \Delta_N$. This requires the adversary to learn the mean μ accurately.

But the mixture distribution is chosen exactly for the purpose of making the adversary's task computationally hard. Indeed, estimating the mean μ based on the samples $\Delta_1, \ldots, \Delta_N$ is a well-known computationally hard problem in robust statistics, called *sparse mean estimation under Huber's contamination noise.*

The choice of the mixture distribution allow us to exploit the following hypothesis testing version of the sparse mean estimation problem.

Sparse Mean Hypothesis Testing:
- $H_0 : \Delta_1, \ldots, \Delta_N$ was sampled from $N(0, \varepsilon^2 I)$.
- $H_1 : \Delta_1, \ldots, \Delta_N$ was sampled from the mixture (4.1).

Clearly, if it is impossible to distinguish H_0 and H_a in the sparse mean hypothesis testing, then it is not possible to estimate accurately the mean μ (or $-\mu$) based on $\Delta_1, \ldots, \Delta_N$.

Perhaps surprisingly, Brennan and Bresler [4] showed that under a well-known conjecture in theoretical computer science, *the k-BPC conjecture*, solving the sparse mean estimation problem can be hard:

Theorem 4.1 (Brennan and Bresler [4])**.** *In the sparsity regime $k \ll \sqrt{d}$, no polynomial-time algorithm on N can solve* (4) *with less than $N = \tilde{\Omega}(k^2)$ samples, assuming the k-BPC conjecture is true. On the other hand, there exists a computationally infeasible algorithm that solves* (4) *with $k = \Theta(k \log d)$ samples.*

The gap between k and k^2 (hiding log factors) is an example of statistical-to-computational gaps, a topic extensively studied in theoretical computer science literature (see [11] and the references therein).

Impossibility results in machine learning, statistics, and computer science, which result in statistical-to-computational gaps, are usually interpreted as "negative" statements. Here, our perspective is different: we leverage a statistical-to-computational gap to our advantage – to safeguard our watermarking from adversarial attacks.

We now prove the main fact about unremovability:

Proposition 4.2. *Assume that $d^\delta \ll N^{1+\delta} \ll d$ for some fixed $\delta > 0$. Let $k = N^{(1+\delta)/2}$. Then, at each step, any Gaussian distribution the adversary can learn in polynomial time has TV distance at least 0.4 from the distribution of the watermarked logits $\ell(i) + \Delta(i)$ in* (4.2)*.*

Proof. The best the adversary can hope for is to predict $\Delta_{adv} \sim N(\mu_{adv}, \varepsilon^2 I)$, for some μ_{adv} satisfying $\|\mu - \mu_{adv}\|_2 \geq 1/2$. (Indeed, if $\|\mu_{adv} - \mu\|_2 < 1/2$, then the adversary would solve the sparse mean estimation hypothesis testing which is not possible in polynomial time thanks to Theorem 4.1. Indeed, to learn the mean μ with higher accuracy one would need more than $\tilde{\Theta}(k^2) \gg N$ samples.)

Notice that the total variation distance between the Δ_{adv} and Δ (or equivalently $-\mu$) is at least $\Phi(-1/4) > 0.4$, where $\Phi(\cdot)$ is the cumulative density function of a standard multivariate Gaussian. $\square$

The reason why we need to assume the regime $d^\delta \ll N^{1+\delta} \ll d$ is to fulfill the hypothesis of the Theorem 4.1 due to the log factors in Theorem 4.1.

Also note that the choice of total variation distance is not essential, we just require a distance between Gaussian distributions that is bounded away from zero if their means are $1/2$-far apart in Euclidean distance. For example, similar guarantees holds for KL-divergence or 2-Wassertein distance.

4.1. **Watermark Detection - Open Source Setting.** We describe our watermarking detection algorithm. The idea is similar to the one we used in the closed setting: if the text is independent from the watermark, then $G_i(x_i)$ are distributed as standard Gaussians and therefore the empirical mean concentrates around 0. On the other hand, if the text is generated by the watermarking scheme then we have a bias towards the positive entries of $G_1, \ldots, G_N$ and consequently the empirical mean should deviate from zero.

We now describe the detection algorithm for the open source setting:

Let c_0 be the absolute constant in [20, Proposition 2.7.1 from (b) to (e)]. Our main result for the open source setting is the following theorem:

Theorem 4.3 (Main Result for the Open Source Setting)**.** *Let c_0 as above. The watermarking scheme described the rule* (4.2) *with $\varepsilon \leq 1/2$ and the detection Algorithm 2 satisfies the following properties:*

(1) It is unremovable.

Moreover, for any $\delta \in (0,1)$,

Algorithm 2 Watermark Detection for the Open Source Setting

Input: The text $x_1, \ldots, x_N$. The key: $G_1, \ldots, G_N$ and ε. Tolerance δ.
Output: True or False.

   $Z \leftarrow \frac{1}{N} \sum_{j=1}^{N} G_j(x_j)$.
   if $Z \geq \varepsilon \sqrt{2 \log(1/\delta)/N}$ **then**
       return: True
   end if
       return: False

---

   (2) The watermarking scheme is sound with probability at least $1 - \delta$.
   (3) Let p_j^* denote the probability of the most likely token in the vocabulary at step $j = 1, \ldots, N$. If, for some $\gamma > 0$,

(4.3)
$$\frac{\varepsilon^2}{1200} \frac{1}{N} \sum_{j=1}^{N} (1 - p_j^*) \geq \varepsilon \sqrt{\frac{2 \log(1/\delta)}{N}} + \gamma.$$

   Then, with probability at least $1 - e^{-\gamma^2 N/672 c_0^2 \varepsilon^2}$, the watermark scheme is complete.

We remark that the assumption (4.3) is analogous to assumption (3.3) and it is somewhat necessary as explained in the Remark 3.3.

We made effort to make all constants explicit, but we opt for a more simplified analysis rather than optimizing the value of the constants. Finally, the assumption on $\varepsilon \leq 1/2$ is for technical convenient and could be replace by any absolute constant if necessary.

Before we proceed to the prove, we require some preliminary results. The proofs of the preliminary results are postponed to the Appendix. Recall that the softmax functions (2.1) and (4.2).

Proposition 4.4. *Let $G = (g_1, \ldots, g_d) \sim N(0, \varepsilon^2 I)$, for some $\varepsilon \leq 1/2$ and $x \in [d]$ be a token sampled according to the watermarked softmax rule (4.2). Set p^* to be the non-increasing rearrangement of the unwatermarked probability distribution p (2.1) of the token x. Then*

$$\mathbb{E}G(x) \geq \frac{\varepsilon^2}{88 e \sqrt{8\pi}} (1 - p^*(1)).$$

Lemma 4.5. *Let $G = (g_1, \ldots, g_d) \sim N(0, \varepsilon^2 I)$, for some $\varepsilon \leq 1/2$ and $x \in [d]$ be a token sampled according to the watermarked softmax rule (4.2). Then the random variable $G(x)$ is sub-exponential with $\|G(x)\|_{\psi_1} \leq 2.8\sqrt{10}\varepsilon$ and $\|G(x) - \mathbb{E}G(x)\|_{\psi_1} \leq 8.4\sqrt{10}\varepsilon$.*

We leave the proofs to the Appendix. To state our main result about the open source setting, let c_B be the absolute constant in the sub-exponential Bernstein's inequality [20, Theorem 2.8.1].

Proof. We already proved the unremovability requirement in Proposition 4.2.

   Soundness: Notice that under the null hypothesis, $x_1, \ldots, x_N$ are independent from $G_1, \ldots, G_N$, therefore the test Z defined in Algorithm 2 is the distributed as $Z \sim N(0, \varepsilon^2/N)$. By the standard estimate for the Gaussian tail,

$$\mathbb{P}\{Z \geq \varepsilon t\} \leq e^{-t^2 N/2},$$

which is at most δ for $t = \sqrt{2\log(1/\delta)/N}$.

Completness: Assume that the text is watermarked. Set $Z_0 := 0$ and notice that

$$Z_k := Z_{k-1} + G_k(x_k) - \mathbb{E}\{G_k(x_k)|x_1,\ldots,x_{k-1}\},$$

is a martingale. By Lemma 4.5, the increments

$$Y_k := [Z_k - Z_{k-1}]|x_1,\ldots,x_{k-1},$$

are sub-exponential. In addition to this, [20, Proposition 2.7.1] implies that there exists a constant $c_0 > 0$ for which all the sub-exponential random variables Y_k satisfy

$$\mathbb{E}e^{\lambda|Y_k|} \leq e^{\lambda^2 c_0^2 \|Y_k\|_{\psi_1}^2} \quad \text{for all} \quad |\lambda| \leq \frac{1}{c_0\|Y_k\|_{\psi_1}}.$$

It follows from the sub-exponential version of the Azuma-Hoeffding's inequality [21, Theorem 2.3] and Proposition 4.4 that

$$\mathbb{P}\left\{Z \geq \frac{\varepsilon^2}{88e\sqrt{8\pi}} \frac{1}{N} \sum_{j=1}^{N}(1 - p_j^*) - t\right\} \geq 1 - e^{-t^2 N/2c_0^2 84\varepsilon^2}.$$

The result follows from $t = \gamma/2$ combined with the assumption (4.3). $\qquad\square$

5. Appendix

5.1. **Proof of Proposition 4.4.**

Proof. To start, we focus on the term

$$\mathbb{E}\left[g_k \frac{e^{\Delta(k)+\ell(k)}}{e^{\ell(1)+\Delta(1)} + \cdots + e^{\ell(d)+\Delta(d)}}\right] \geq e^{-1/\sqrt{k}}\mathbb{E}\left[g_k \frac{e^{g_k}}{e^{\ell(1)+g_1-\ell(k)} + \cdots + e^{\ell(d)+g_d-\ell(k)}}\right].$$

Clearly, $e^{-1/\sqrt{k}} \geq e^{-1}$ as $k \geq 1$. Next, denote by $\mathbb{E}_j$ the expectation with respect to the randomness of g_j only. By the iterated law of expectation, we compute the expectation term in the right-hand side by

$$\mathbb{E}_{1,\ldots,k-1,k+1,\ldots,d}\left[\mathbb{E}_k\left[g_k \frac{e^{g_k}}{e^{\ell(1)+g_1-\ell(k)} + \cdots + e^{\ell(d)+g_d-\ell(k)}}\right]\right],$$

and by independence we may treat $\sum_{i\neq k} e^{\ell(i)-\ell(k)+g_i}$ as a constant for the inner expectation. Thus, let us define $\alpha_k := \sum_{i\neq k} e^{\ell(i)-\ell(k)+g_i}$ and write

$$\mathbb{E}_k\left[g_k \frac{e^{g_k}}{\alpha_k + e^{g_k}}\right] = \frac{1}{2}\mathbb{E}_k\left[|g_k| \frac{e^{|g_k|}}{\alpha_k + e^{|g_k|}} - |g_k| \frac{e^{-|g_k|}}{\alpha_k + e^{-|g_k|}}\right]$$

$$= \frac{\alpha_k}{2}\mathbb{E}_k\left[|g_k|\left(\frac{e^{|g_k|} - e^{-|g_k|}}{\alpha_k^2 + \alpha_k(e^{|g_k|} + e^{-|g_k|}) + 1}\right)\right].$$

Next, notice that the function $f : [0,\infty) \to \mathbb{R}$

$$f(x) := \frac{e^x - e^{-x}}{\alpha_k^2 + \alpha_k(e^x + e^{-x}) + 1},$$

is increasing and non-negative. Invoking the FKG inequality, we have that

$$\mathbb{E}_k\left[g_k\frac{e^{g_k}}{\alpha_k+e^{g_k}}\right] \geq \frac{\varepsilon\alpha_k}{\sqrt{2\pi}}\mathbb{E}_k\left[\frac{e^{|g_k|}-e^{-|g_k|}}{\alpha_k^2+\alpha_k(e^{|g_k|}+e^{-|g_k|})+1}\right]$$

$$\geq \frac{\varepsilon\alpha_k}{\sqrt{2\pi}}\left(\frac{e^{\varepsilon c}-e^{-\varepsilon c}}{\alpha_k^2+\alpha_k(e^{\varepsilon c}+e^{-\varepsilon c})+1}\right)\mathbb{P}\{|g_k|\geq\varepsilon c\}.$$

By the standard tail estimate for Gaussians, for the choice of $c=1/2$, it follows that $\mathbb{P}\{|g_k|\geq\varepsilon/2\}\geq 1/2$. Recall that $\varepsilon\leq 1/2$ which implies that $e^{\varepsilon/2}+e^{-\varepsilon/2}\leq 2.1$, thus

$$\alpha_k^2+\alpha_k(e^{\varepsilon/2}+e^{-\varepsilon/2})+1 \leq \frac{2.06}{2}\left(\alpha_k^2+2\alpha_k+1\right) \leq 1.05\left(\alpha_k^2+2\alpha_k+1\right),$$

and

$$\mathbb{E}_k g_k\left[\frac{e^{g_k}}{\alpha_k+e^{g_k}}\right] \geq \frac{\varepsilon\alpha_k}{\sqrt{8\pi}}\left(\frac{e^{\varepsilon/2}-e^{-\varepsilon/2}}{\alpha_k^2+\alpha_k(e^{\varepsilon/2}+e^{-\varepsilon/2})+1}\right)$$

$$\geq \varepsilon\frac{e^{\varepsilon/2}-e^{-\varepsilon/2}}{1.05\sqrt{8\pi}}\left(\frac{\alpha_k}{\alpha_k^2+2\alpha_k+1}\right)$$

$$\geq \frac{\varepsilon^2}{1.05\sqrt{8\pi}}\left(\frac{\alpha_k}{\alpha_k^2+2\alpha_k+1}\right) = \frac{\varepsilon^2}{1.05\sqrt{8\pi}}\frac{\alpha_k}{(\alpha_k+1)^2}.$$

It remains to estimate (from below)

$$(5.1) \qquad \frac{\varepsilon^2}{1.05\sqrt{8\pi}}\sum_{k=1}^{d}\mathbb{E}_{1,\ldots,k-1,k+1,\ldots,d}\left[\frac{\alpha_k}{(\alpha_k+1)^2}\right].$$

Or equivalently (up to the multiplicative constant in front),

$$\sum_{k=1}^{d}\mathbb{E}_{1,\ldots,k-1,k+1,\ldots,d}\left[\frac{1}{(\alpha_k+1)}-\frac{1}{(\alpha_k+1)^2}\right].$$

Recalling that $\alpha_k=\sum_{i\neq k}e^{\ell(i)-\ell(k)+g_i}$, suppose that there is a non-empty event $\mathcal{E}$ for which both conditions below hold simultaneously
(5.2)
$$\sum_{i\neq k}e^{\ell(i)-\ell(k)+g_i} \leq 4.55\sum_{i\neq k}e^{\ell(i)-\ell(k)} \quad\text{and}\quad \sum_{i\neq k}e^{\ell(i)-\ell(k)+g_i} \geq \frac{1}{4}\sum_{i\neq k}e^{\ell(i)-\ell(k)}.$$

By the second estimate in (5.2), we have that

$$\frac{1}{\alpha_k} = \frac{e^{\ell(k)}}{\sum_{i\neq k}e^{\ell(i)+g_i}} \leq 4\frac{e^{\ell(k)}}{\sum_{i\neq k}e^{\ell(i)}} = \frac{4p(k)}{1-p(k)},$$

and then

$$\frac{1}{(1+\alpha_k)^2} \leq \frac{1}{(1+\alpha_k)(1+(1-p(k))/4p(k))}.$$

Consequently,

$$\frac{1}{1+\alpha_k}-\frac{1}{(1+\alpha_k)^2} \geq \left(\frac{1}{1+\alpha_k}\right)\frac{1-p(k)}{4p(k)+1-p(k)} \geq \left(\frac{1}{1+\alpha_k}\right)\frac{1-p^*(1)}{4}.$$

Finally, notice that the first estimate in (5.2) implies that

$$\sum_{k=1}^{d} \frac{1}{1 + \alpha_k} = \sum_{k=1}^{d} \frac{e^{\ell(k)}}{e^{\ell(k)} + 4.55 \sum_{i \neq k} e^{\ell(i)}}$$

$$\geq \frac{1}{4.55} \sum_{k=1}^{d} \frac{e^{\ell(k)}}{e^{\ell(k)} + \sum_{i \neq k} e^{\ell(i)}} = \frac{1}{4.55} \geq 0.2.$$

Since (5.1) is non-negative, we have that

$$\sum_{k=1}^{d} \mathbb{E}_{1,\ldots,k-1,k+1,\ldots,d} \left[\frac{\alpha_k}{(\alpha_k + 1)^2} \right] \geq 0.05(1 - p^*(1))\mathbb{P}\{\mathcal{E}\}$$

All it remains is to prove that $\mathbb{P}\{\mathcal{E}\}$ is bounded away from zero. To this end, notice that by Markov's inequality

$$\mathbb{P}\{\alpha_k \leq 4\mathbb{E}\alpha_k\} \geq \frac{3}{4}.$$

Next, notice that for every $i \neq k$,

$$\frac{e^{\ell(i)}}{\sum_{i \neq k} e^{\ell(i)}} e^{g_i} \geq \frac{e^{\ell(i)}}{\sum_{i \neq k} e^{\ell(i)}} \mathbb{1}_{g_i \geq 0} =: a_i \mathbb{1}_{g_i \geq 0}.$$

The random variable $Y := \sum_{i \neq k} a_i \mathbb{1}_{g_i \geq 0}$ is the sum of independent non-negative random variables Y_i, where $Y_i \in [0, a_i]$. Notice that $\sum_{i \neq k} a_i = 1$, thus we split into two cases. If there exists an $a_i \geq 1/4$ then with probability $1/2$, $Y \geq 1/4$. On the other hand, if $a_i \leq 1/4$ for all $i \neq k$ then by Hoeffding's inequality

$$\mathbb{P}\left\{ Y \geq \frac{1}{2} - t \right\} \geq 1 - e^{-32t^2}.$$

Setting $t = 1/4$, we obtain that

$$\mathbb{P}\left\{ Y \geq \frac{1}{4} \right\} \geq 1 - e^{-2}.$$

We conclude that $\mathbb{P}(\mathcal{E}) \geq \min\{1/2 - e^{-2}, 3/4 - 1/2\} = 1/4$, which finishes the proof. $\square$

5.2. **Proof of Lemma 4.5.**

Proof. Since $\|\cdot\|_{\psi_1}$ is a norm, we have that

$$\|G(x) - \mathbb{E}G(x)\|_{\psi_1} \leq \|G(x)\|_{\psi_1} + \|\mathbb{E}G(x)\|_{\psi_1} = \|G(x)\|_{\psi_1} + |\mathbb{E}G(x)| \leq 3\|G(x)\|_{\psi_1},$$

where the last step follows from

$$\left| \frac{1}{\|G(x)\|_{\psi_1}} \mathbb{E}G(x) \right| \leq \left| 1 + \frac{1}{\|G(x)\|_{\psi_1}} \mathbb{E}G(x) \right| \leq \mathbb{E}e^{|G(x)|/\|G(x)\|_{\psi_1}} \leq 2.$$

The proof boils down to showing that for some well-chosen τ (small as possible)

$$\mathbb{E}e^{|G(x)|/\tau} \leq e^{1/\sqrt{k}} \sum_{k=1}^{d} \mathbb{E}\left[e^{g_k/\tau} \frac{e^{g_k}}{e^{\ell(1)-\ell(k)+g_1} + \ldots + e^{\ell(d)-\ell(k)+g_d}} \right] \leq 2,$$

which implies that $\|G(x)\|_{\psi_1} \le \tau$. We proceed similarly as in Proposition 4.4. Let $\alpha_k := \sum_{i \ne k} e^{\ell(i)-\ell(k)+g_i}$ and notice that by Cauchy-Schwarz inequality,

$$\sum_{k=1}^{d} \mathbb{E}_k \left[e^{g_k/\tau} \frac{e^{g_k}}{e^{\ell(1)-\ell(k)+g_1} + \ldots + e^{\ell(d)-\ell(k)+g_d}} \right]$$
$$\le \sum_{i=1}^{d} (\mathbb{E}_k e^{2g_k/\tau})^{1/2} \left(\mathbb{E}_k \left[\frac{e^{2g_k}}{(\alpha_k + e^{g_k})^2} \right] \right)^{1/2} .$$

We claim that

$$\left(\mathbb{E}_k \left[\frac{e^{g_k}}{(\alpha_k + e^{g_k})^2} \right] \right)^{1/2} \le 2.2 \mathbb{E}_k \left[\frac{e^{\ell(k)+g_k}}{e^{\ell(1)+g_1} + \ldots + e^{\ell(d)+g_d}} \right].$$

If the claim is true then by the law of iterated expectation

$$\mathbb{E} e^{|G(x)|/\tau} \le 2.2 e (\mathbb{E}^{2g/\tau})^{1/2} \mathbb{E} \left[\sum_{k=1}^{d} \frac{e^{\ell(k)+g_k}}{e^{\ell(1)+g_1} + \ldots + e^{\ell(d)+g_d}} \right] \le 6 e^{\varepsilon^2/\tau^2}.$$

Choosing $\tau \ge \sqrt{10}\varepsilon$, we reach the estimate $\mathbb{E} e^{|G(x)|/\tau} \le 6.7$. We would like to replace the constant 6.7 by 2 in the right-hand side. To this end, notice that for the constant $a = \log 6.7/\log 2 > 1$, the function $f(x) = x^{1/a}$ is concave and then Jensen inequality implies that

$$\mathbb{E} e^{|G(x)|/a\tau} = \mathbb{E} \left[(e^{|G(x)|/\tau})^{1/a} \right] \le \left(\mathbb{E} e^{|G(x)|/\tau} \right)^{1/a} \le (6.7)^{1/a} = 2.$$

Thus setting $\tau = 2.8\sqrt{10}\varepsilon$ concludes the proof. We now proceed to prove the claim. At one hand,

(5.3)
$$\mathbb{E} \left[\frac{e^{g_k}}{\alpha_k + e^{g_k}} \right] = \frac{1}{2} \mathbb{E} \left[\frac{e^{-|g_k|}}{\alpha_k + e^{-|g_k|}} \right] + \frac{1}{2} \left[\frac{e^{|g_k|}}{\alpha_k + e^{|g_k|}} \right]$$
$$\ge \frac{1}{\alpha_k + 1} \left(\frac{1}{2} \mathbb{E} e^{-|g_k|} + \frac{1}{2} \right)$$
$$\ge \frac{1}{\alpha_k + 1} \left(\frac{0.95}{2} e^{-2\varepsilon} + \frac{1}{2} \right) \quad (\text{as } \mathbb{P}\{|g_k| \ge 2\varepsilon\} \ge 0.95)$$
$$\ge \frac{0.67}{\alpha_k + 1} \quad (\text{as } \varepsilon \le 1/2).$$

On the other hand,

$$
\begin{aligned}
\mathbb{E}\left[\frac{e^{2g_k}}{(\alpha_k + e^{g_k})^2}\right] &= \frac{1}{2}\left(\mathbb{E}\left[\frac{e^{2|g_k|}}{(\alpha_k + e^{|g_k|})^2}\right] + \mathbb{E}\left[\frac{e^{-2|g_k|}}{(\alpha_k + e^{-|g_k|})^2}\right]\right) \\
&\leq \frac{1}{2}\left(2e^{2\varepsilon^2}\frac{1}{(\alpha_k + 1)^2} + \mathbb{E}\left[\frac{e^{-2|g_k|}}{(\alpha_k + e^{-|g_k|})^2}\right]\right) \\
&= \frac{1}{2}\left(2e^{2\varepsilon^2}\frac{1}{(\alpha_k + 1)^2} + \mathbb{E}\left[\frac{1}{(\alpha_k e^{|g_k|} + 1)^2}\right]\right) \\
&= \left(e^{2\varepsilon^2}\frac{1}{(\alpha_k + 1)^2} + \mathbb{E}\left[\frac{1}{2(\alpha_k e^{|g_k|} + 1)^2}\right]\right) \\
&= \frac{1}{(\alpha_k + 1)^2}\left(e^{2\varepsilon^2} + \frac{1}{2}\right) \\
&\leq \frac{1}{(\alpha_k + 1)^2}(\sqrt{e} + \frac{1}{2}) \quad (\text{as } \varepsilon \leq 1/2).
\end{aligned}
$$
(5.4)

Putting together (5.4) and (5.3) and the fact that $\varepsilon < 1/2$, we obtain that

$$
\left(\mathbb{E}\left[\frac{e^{2g_k}}{(\alpha_k + e^{g_k})^2}\right]\right)^{1/2} \leq 2.18\mathbb{E}\left[\frac{e^{g_k}}{\alpha_k + e^{g_k}}\right] = 2.18\mathbb{E}\left[\frac{e^{\ell(k)+g_k}}{e^{\ell(1)+g_1} + \ldots + e^{\ell(d)+g_d}}\right].
$$

$\square$

ACKNOWLEDGMENTS

REFERENCES

[1] Scott Aaronson and H Kirchner. Watermarking of large language models. In *Large Language Models and Transformers Workshop at Simons Institute for the Theory of Computing*, 2023.

[2] Sahar Abdelnabi and Mario Fritz. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 121–140. IEEE, 2021.

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

[4] Matthew Brennan and Guy Bresler. Reducibility and statistical-computational gaps from secret leakage. In *Conference on Learning Theory*, pages 648–847. PMLR, 2020.

[5] Miranda Christ, Sam Gunn, Tal Malkin, and Mariana Raykova. Provably robust watermarks for open-source language models. *arXiv preprint arXiv:2410.18861*, 2024.

[6] Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1125–1139. PMLR, 2024.

[7] Geoffrey A Fowler. We tested a new chatgpt-detector for teachers. it flagged an innocent student. *The Washington Post*, 3, 2023.

[8] Peter J Huber and Elvezio M Ronchetti. *Robust statistics*. John Wiley & Sons, 2011.

[9] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR, 2023.

[10] Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

[11] Dmitriy Kunisky, Alexander S Wein, and Afonso S Bandeira. Notes on computational hardness of hypothesis testing: Predictions using the low-degree likelihood ratio. In *ISAAC Congress (International Society for Analysis, its Applications and Computation)*, pages 1–50. Springer, 2019.

[12] Xiang Li, Feng Ruan, Huiyuan Wang, Qi Long, and Weijie J Su. A statistical framework of watermarks for large language models: Pivot, detection efficiency and optimal rules. *The Annals of Statistics*, 53(1):322–351, 2025.

[13] Silvia Milano, Joshua A McGrane, and Sabina Leonelli. Large language models challenge the future of higher education. *Nature Machine Intelligence*, 5(4):333–334, 2023.

[14] Travis Munyer and Xin Zhong. Deeptextmark: Deep learning based text watermarking for detection of large language model generated text. *arXiv e-prints*, pages arXiv–2305, 2023.

[15] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[16] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

[17] Kate Starbird. Disinformation's spread: bots, trolls and all of us. *Nature*, 571(7766):449–450, 2019.

[18] Chris Stokel-Walker. Ai bot chatgpt writes smart essays-should professors worry? *Nature*, 2022.

[19] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Open and efficient foundation language models. *Preprint at arXiv. https://doi. org/10.48550/arXiv*, 2302(3), 2023.

[20] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[21] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

[22] KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. Robust natural language watermarking through invariant features. *arXiv preprint arXiv:2305.01904*, 4, 2023.

[23] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.