# WatME: Towards Lossless Watermarking Through Lexical Redundancy

**Liang Chen**[♠] **Yatao Bian**[♡] **Yang Deng**[♢] **Deng Cai**[♡]
**Shuaiyi Li**[♠] **Peilin Zhao**[♡] **Kam-Fai Wong**[♠]
[♠] The Chinese University of Hong Kong [♡] Tencent AI Lab [♢] National University of Singapore
`{lchen, kfwong}@se.cuhk.hk`

## Abstract

Text watermarking has emerged as a pivotal technique for identifying machine-generated text. However, existing methods often rely on arbitrary vocabulary partitioning during decoding to embed watermarks, which compromises the availability of suitable tokens and significantly degrades the quality of responses. This study assesses the impact of watermarking on different capabilities of large language models (LLMs) from a cognitive science lens. Our finding highlights a significant disparity; knowledge recall and logical reasoning are more adversely affected than language generation. These results suggest a more profound effect of watermarking on LLMs than previously understood. To address these challenges, we introduce Watermarking with Mutual Exclusion (WatME), a novel approach leveraging linguistic prior knowledge of inherent lexical redundancy in LLM vocabularies to seamlessly integrate watermarks. Specifically, WatME dynamically optimizes token usage during the decoding process by applying a mutually exclusive rule to the identified lexical redundancies. This strategy effectively prevents the unavailability of appropriate tokens and preserves the expressive power of LLMs. We provide both theoretical analysis and empirical evidence showing that WatME effectively preserves the diverse capabilities of LLMs while ensuring watermark detectability. Our code will be released to facilitate future research. via `https://github.com/ChanLiang/WatME`.

## 1 Introduction

The advent of large language models (Ouyang et al., 2022; OpenAI, 2023a) with human-level generative capabilities presents tremendous opportunities across diverse domains (Deng et al., 2023; Li et al., 2024; Wang et al., 2023). However, their ability to synthesize high-quality text also raises widespread concerns about potential misuse, including the dissemination of misinformation (Zellers et al., 2019; Chen et al., 2023a) and the facilitation of academic dishonesty (Stokel-Walker, 2022). This necessitates developing techniques to reliably attribute generated text to AI systems.

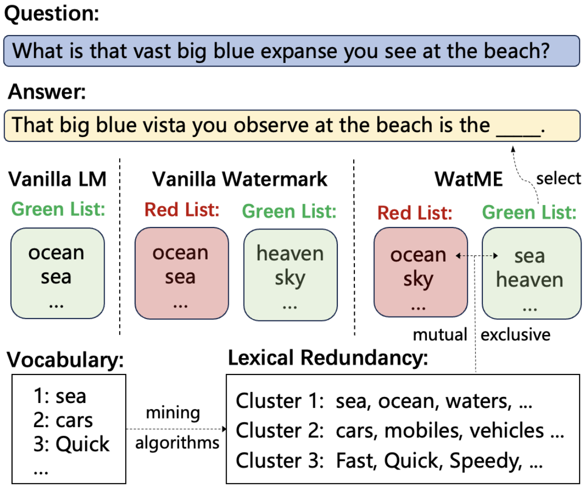


Figure 1: An illustration of WatME's advantage for lossless watermarking. The left panel depicts a vanilla LM with all words available during generation. The middle panel exposes the flaw in vanilla watermarking, which may assign all suitable tokens (e.g., 'ocean' and 'sea') to the red list, diminishing text quality. The right panel underlines how WatME exploits lexical redundancy by applying a mutual exclusion rule between such words, ensuring at least one suitable word remains on the green list, thereby improving text quality.

Existing approaches typically fall into two main paradigms. The first type attempts to distinguish machine-generated text by hunting for inductive statistical or linguistic patterns (Gehrmann et al., 2019; Mitchell et al., 2023; Zellers et al., 2019; OpenAI, 2023b), employing methods that span from basic manual feature engineering to the intricate training of complex classifiers. However, as generative models continue improving, their outputs increasingly resemble the pattern of human writing, rendering statistical detectors ineffective (Dou et al., 2022; Sadasivan et al., 2023; Chakraborty et al., 2023). The second paradigm

*Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9166–9180
August 11-16, 2024 ©2024 Association for Computational Linguistics

promotes a more proactive approach, advocating for direct intervention in the generative process to actively watermark model outputs (Kirchenbauer et al., 2023; Christ et al., 2023; Zhao et al., 2023). This strategy embeds identifiable fingerprints within machine-generated text, enabling provenance verification. As LLMs' capabilities continue to grow, this approach is more effective in detecting LLM-generated text (Sadasivan et al., 2023). However, introducing watermarks during text generation can significantly impact output quality, which has been a consistent challenge for model developers - how to effectively watermark while preserving text quality.

Recent studies have attempted to improve text quality by ensuring unbiased output distributions in watermarking (Kuditipudi et al., 2023; Hu et al., 2024), while employing pseudorandomness-guided perturbations or reweighting to adjust the original output distributions of LLMs. However, an unbiased distribution in expectation does not guarantee high text quality, and the use of these techniques reduces the effectiveness of watermark detection, especially in models that have undergone alignment training (Kuditipudi et al., 2023), thereby diminishing the practical utility of these methods.

In this paper, we introduce a novel approach to text watermarking by leveraging engineered lexical redundancy during the decoding phase of language generation. Our method utilizes the comprehensive set of tokens available to a language model, clustering them based on overlapping semantic or syntactic functionalities to create sets of interchangeable tokens. This process simulates redundancy within the lexical space, akin to the surplus pixels in images that facilitate watermarking in multimodal data (Nikolaidis and Pitas, 1999; Samuel and Penzhorn, 2004). The motivation for this strategy arises from the challenge of applying traditional watermarking techniques to textual data. In contrast to the inherent redundancy found in images, the discrete and succinct nature of textual data offers little to no native redundancy, making it challenging to exploit redundancy in the textual space (Zhou et al., 2021; He et al., 2022). By engineering lexical redundancy, our method not only surmounts the limitations imposed by the inherent properties of natural language but also paves the way for secure and efficient text watermarking.

After exploring these redundancies, we exploit them via our novel algorithm, WatME, which enhances text quality by integrating a mutual exclusivity rule within the context of lexical redundancy during the watermarking process. Specifically, WatME refines the decoding process by explicitly assigning words within each redundant cluster to distinct 'green' or 'red' teams, ensuring that no single cluster is wholly allocated to one team. Our approach confers two main advantages: (1) it enables the 'green' team to capture a broader array of semantics, thereby boosting the model's expressive power; and (2) it increases the probability that the LLM selects the most appropriate word at each decoding step, e.g., in Figure 1, vanilla watermarking can assign all suitable words to the 'red' list, thus severely impairing performance. In contrast, our approach guarantees the presence of at least one appropriate word, thus preserving the model's expressiveness. Building on these methodological advances, extensive theoretical and empirical evidence supports their effectiveness without compromising detection capabilities. These improvements significantly bolster the emergent abilities of large models under watermarks, surpassing the performance of baseline methods.

Our main contributions are as follows:

- Motivated by multimedia data's inherent redundancy and the precise conciseness of text, we propose two distinct approaches for mining ***lexical redundancy***.
- We develop the WatME algorithm, which embeds mutual exclusion rules within the lexical space for text watermarking. Theoretical analysis is presented to validate its effectiveness in preserving the quality of text responses.
- Experimental results show that WatME effectively outperforms existing methods in retaining the emergent capabilities of LLMs, notably knowledge recall and logical reasoning, within the conceptual framework of Cattell's cognitive theory, without compromising detectability.

## 2 Related Work

Early works on AI-generated text detection develop post-hoc detection methods to analyze machine-generated text by treating the problem as a binary classification task (OpenAI, 2019; Jawahar et al., 2020; Mitchell et al., 2023). For instance, OpenAI has fine-tuned RoBERTa (Liu et al., 2019) to distinguish between human and GPT-2 generated texts (OpenAI, 2019). However, existing detectors are found to be fragile against adversarial attacks (Wolff, 2020) and biased towards non-native English writers (Liang et al., 2023). Moreover, as LLMs continue to advance, their generated outputs more closely resemble human-written text, rendering these methods progressively less effective.

On the other side, watermarking, traditionally a copyright marking method (Adi et al., 2018; Rouhani et al., 2018), involves developers, users, and regulatory entities. Developers choose an algorithm to subtly embed hidden modifications into data, which can be altered during user transmission. Regulatory bodies can later extract this information to trace and regulate AI-generated content (Atallah et al., 2001; Wilson et al., 2014; Hacker et al., 2023). In the context of natural languages, watermarking typically involves modifying content or structure. For example, rule-based methods (Stefan et al., 2000) or carefully designed neural encoders (Yang et al., 2022; Ueoka et al., 2021) encrypt messages into text, which are then extracted using the corresponding rules and neural decoder. The discrete nature of natural language, however, presents a considerable challenge to this approach, as modifications can unintentionally degrade text quality or alter its intended meaning.

For the detection of LLM-generated texts, a pioneering watermarking technique (Kirchenbauer et al., 2023) partitions tokens into 'green' and 'red' lists, biases output distribution towards 'green' tokens, and creates patterns that are detectable yet imperceptible to humans. Nevertheless, while yielding promising detection results, these methods may still degrade the textual quality and be vulnerable to the paraphrase attack. Current efforts (Christ et al., 2023; Fernandez et al., 2023; Zhao et al., 2023) in this field aim to develop more robust watermarking methods capable of defending various user attacks.

Apart from improving robustness, a few studies have recognized the importance of enhancing the quality of text produced by watermarked LLMs. (Kuditipudi et al., 2023) utilizes Gumbel softmax to incorporate pseudorandomdness-based randomness into the output distribution of language models. While this technique alters the probability distribution, the Gumbel softmax ensures that the expected distribution remains approximately unchanged, thus rendering the watermarking process unbiased. Recent work (Hu et al., 2024) also shares a similar philosophy of employing reweighting technology for unbiased output distribution transformations, preserving the expected distribution unbiased. However, unbiased distribution can not guarantee unaffected text quality. Furthermore, these methodologies have shown a marked decrease in detection performance, particularly for aligned LLMs (Kuditipudi et al., 2023). Addressing these shortcomings, our research introduces a novel paradigm that exploits the intrinsic redundancy in the text generation process of LLMs to create more lossless watermarks, with a special emphasis on LLMs' emergent capabilities, thereby offering a watermarking solution that is both lossless and consistently detectable.

## 3 Method

In this section, we begin by providing a summary of the preliminaries related to text watermarking. Subsequently, we delve into an investigation of redundancy in the lexical space and demonstrate how this redundancy can be leveraged to develop a watermarking algorithm that achieves a higher degree of losslessness for large language models. Finally, we employ mathematical analysis to elucidate the benefits of our proposed method.

### 3.1 Preliminary

The watermarking process is composed of two fundamental procedures: watermark encoding and watermark detection. The encoding procedure is carried out by developers to insert a watermark into an output natural language sequence $\boldsymbol{y}$, generated by a LLM $\mathcal{M}$ for a given prompt $\boldsymbol{x}$. While the detection procedure, performed by regulators, involves the extraction and identification of the watermark from the sequence $\boldsymbol{y}$ for the purpose of monitoring the output from model $\mathcal{M}$. The algorithms that detail these procedures are described in the Appendix A.

The watermark encoding process is guided by two parameters: $\gamma$ and $\delta$. At each decoding step $t$, it uses a hash key, which could be the index of the previous token, to partition the vocabulary $\mathcal{V}$ into two subsets: a green list $G_t$ which encourages usage, and a red list $R_t$ which discourages usage. The parameter $\gamma$ determines the size of the green list, while $\delta$ specifies the degree of encouragement for the green list, the increase in current logits $\boldsymbol{\ell}_t$ before performing softmax, as Eq.1. As $\delta$ rises, the watermark becomes more detectable in the subsequent detection process, but it may also compromise the quality of the generation. In real-world regulatory scenarios, where high detectability is required, a large $\delta$ value is generally preferred.

$$\begin{aligned} \hat{\boldsymbol{\ell}}_t[i] &:= \boldsymbol{\ell}_t[i] + \delta, \qquad i \in G_t \\ \hat{\boldsymbol{p}}_t &= softmax(\hat{\boldsymbol{\ell}}_t) \end{aligned} \tag{1}$$

The watermark detection process counts the number of green list tokens within $\boldsymbol{y}$, denoted by $|\boldsymbol{y}|_G$, using Eq.2. This process begins with the null hypothesis $H_0$: *The text sequence is generated without adherence to the green list rule*. A $z$-statistic is then computed by Eq.3. If the $z$-score surpasses a pre-specified threshold, the null hypothesis is rejected, and the watermark is identified.

$$|\boldsymbol{y}|_G = \sum\nolimits_{t=1}^{n} \mathbb{1}(y_t \in G_t), \tag{2}$$

$$z_{\boldsymbol{y}} = (|\boldsymbol{y}|_G - \gamma|\mathcal{V}|) \,/\, \sqrt{|\mathcal{V}|\gamma(1-\gamma)}. \tag{3}$$

### 3.2 Explore the Redundancy in Lexical Space

**Concept of Lexical Redundancy** Inspired by the success of image watermarking, we hypothesize that identifying redundancy within data can enable watermarking that doesn't compromise text quality. We thus explore the same opportunities within textual data, a challenging task given the discrete nature of natural language.

To address this challenge, we introduce a related concept in NLP: ***lexical redundancy***. This phenomenon arises during text generation when the most appropriate word is selected from a large, preconstructed vocabulary. Often, this vast vocabulary includes numerous words with similar semantic and syntactic functions — a feature that makes these words interchangeable, thereby resulting in the inherent redundancy in the lexical space.

Our interest in exploring lexical redundancy is grounded in the understanding that interchangeable synonyms, even when used in varied contexts, can deliver similar or identical semantic or syntactic functions. This insight assists in preserving the quality of text generation through an optimized watermark encoding method. For instance, if a suitable word is allocated to the red list, while its synonym is placed in the green list, then the language model can still express the intended semantics or accomplish the necessary syntactic functions. This understanding forms the primary motivation for investigating lexical redundancy.

**Constructing Redundant Lexical Clusters** To this end, we now focus on the construction of lexical redundancy. This process involves automatically grouping tokens—each with similar semantic or syntactic functions—from the language model's vocabulary into clusters. Each cluster, made up of interchangeable tokens, is designed to express a specific semantic or syntactic unit.

To obtain high-quality redundant lexical clusters, we propose the following two different methods: the dictionary-based method, and the prompting-based method:

- **Dictionary-Based Method:** Utilize external dictionaries, such as WordNet (Miller, 1992) and Youdao Dictionary, to discover synonyms within the vocabulary. These synonyms often can be substituted for each other, although there are inevitably some cases where they cannot be interchanged due to polysemy. This method is beneficial for exploiting established synonym relationships but is limited to complete words due to its dependency on external resources.

- **Prompting-based Method:** We prompt large language models, such as LLaMA2 (Touvron et al., 2023), to infer synonyms for a given token by utilizing in-context learning techniques (Brown et al., 2020a), with the demonstrations being annotated manually by us. Detailed prompts are deferred to Appendix B.

To acquire higher-quality clusters with fully interchangeable tokens, we employed two strategies during the mining process:

**Handling Subword Tokenization** Subword tokenization blends word and character-based approaches (Sennrich et al., 2016; Schuster and Nakajima, 2012; Kudo and Richardson, 2018), challenges the mining of redundant lexical clusters in neural text processing. This technique typically retains common words as full units and decomposes rare words into subunits. Our research mitigates these challenges by ***concentrating on intact, frequently used words during preprocessing***, thereby diminishing noise and simplifying the algorithm.

**Incorporating Grammatical Factors** In the context of English, the identification of interchangeable words demands consideration of grammatical factors—tense, voice, and number—alongside semantic similarity. For instance, 'car' and 'vehicles' differ in number, affecting interchangeability. Our method addresses these issues by implementing a rule set that screens for grammatical inconsistencies, ensuring the generation of coherent and high-quality lexical clusters for subsequent use.

These strategies yield lexical clusters, with each row in Figure 1's bottom right panel representing a cluster of interchangeable tokens. Cluster quality is manually evaluated in Section 6.1.

### 3.3 WatME: Exploit the Lexical Redundancy

Having constructed redundant clusters within the lexical space, we now turn to exploit these for a lossless watermark algorithm.

To facilitate the description of our algorithm, we provide some definitions: A subset $S \subseteq \mathcal{V}$ is defined within the vocabulary $\mathcal{V}$ of a language model $\mathcal{M}$. This subset specifically comprises complete tokens that share synonyms within the vocabulary. We then denote a collection of redundant lexical clusters we mined as $C = \{C_i \mid i = 1..n\}$ such that $\bigcup_{i=1}^{n} C_i = S$. Each cluster, $C_i$, is represented as a token collection $C_i = \{s_{ij} \mid j = 1..m_i, s_{ij} \in S\}$ for $i = 1..n$, and any pair of tokens $s_{ij}, s_{ik} \in C_i$ are interchangeable. We propose to implement our understanding of lossless watermarks by introducing ***a mutual exclusion rule*** building on the identified lexical clusters: interchangeable tokens are mutually exclusive during partitioning. In other words, if a fraction of tokens $\mathcal{A}$, representing a certain semantic or syntactic function, is assigned to the red list, then their remaining synonyms $\mathcal{B}$ should be placed on the green list, and vice versa.

We then detail the WatME encoding process, outlined in Alg. 1, which employs a two-step partitioning process to form a green and red list. The first partition occurs within the redundant lexical clusters $C$ that we have identified within $S$, while the second takes place among the remaining part in the vocabulary denoted as $\mathcal{V} \setminus \mathcal{S}$. We use $\gamma$ to determine the number of tokens from the mined clusters that are allocated to the green list $G'_t$ in the first partition. The remaining tokens, based on the principle of mutual exclusivity, are assigned to the red team $R'_t$. The second partition continues to allocate words to the green list $G_t$ from the remaining vocabulary until the combined size of the green teams from both steps reaches the predefined limit, $\gamma$. The rest of the process follows the steps outlined in the vanilla watermarking of Alg. 2.

The WatME detection algorithm is unchanged, except the green list calculation now uses Alg. 2.

### 3.4 Theoretical Analysis

We provide a mathematical analysis demonstrating how WatME outperforms the conventional method, focusing on the 'green' team's expressiveness and the probability of high-quality sampling.

**Algorithm 1** WatME Encoding

```
Input: prompt $x_1 \cdots x_m$, green list size $\gamma \in (0, 1)$, watermark strength $\delta > 0$.
for $t = 0, 1, \cdots, T-1$ do
    1. Get the logit $\boldsymbol{\ell}_t \in \mathbb{R}^{|\mathcal{V}|}$ from $\mathcal{M}$.
    2. Use seed from the last token, split each cluster $C_i$ in parallel into green list $G'_{it}$ (of size $|C_i|\gamma$) and red list $R'_{it}$ (of size $(1-\gamma)|C_i|$) . Let $G'_t = \cup_i G'_{it}$ and $R'_t = \cup_i R'_{it}$.
    3. Partition the remaining part $\mathcal{V} \setminus \mathcal{S}$ into a green list $G_t$ of size $\gamma|V| - |G'_t|$ and a red list $R_t$ of size $(1-\gamma)|V| - |R'_t|$.
    4. Merge lists from the previous two steps: $G_t = G_t \cup G'_t$ and $R_t = R_t \cup R'_t$.
    5. Add $\delta$ to the elements of logit $\boldsymbol{\ell}_t$ corresponding to the green list, then softmax.
           $\hat{\boldsymbol{p}}_t = softmax(\boldsymbol{\ell}_t[i] + \delta), i \in G_t$
    6. Sample the next token $y_{t+1}$ from $\hat{\boldsymbol{p}}_t$.
end for
Output: watermarked text $y_1 \cdots y_T$.
```

**Definition 3.1 (Semantic Entropy)** *Let $\mathcal{V}$ represent the vocabulary of a language model. We define the semantic entropy of $\mathcal{V}$, denoted by $H_{sem}(\mathcal{V})$, as the entropy of the semantic distribution across $\mathcal{V}$. This entropy quantifies the diversity and richness of meanings expressible by $\mathcal{V}$. Consequently, a higher value of $H_{sem}(\mathcal{V})$ signifies a vocabulary with greater semantic richness.*

**Definition 3.2 (Intrinsic Expressiveness)** *It is assumed that a language model $\mathcal{M}$, with a vocabulary $\mathcal{V}$ characterized by high semantic entropy as indicated by $H_{sem}(\mathcal{V})$, possesses an enhanced intrinsic expressive capacity. This capacity is unaffected by the output distribution of $\mathcal{M}$ and is due to the extensive semantic capabilities of $\mathcal{V}$, which endow $\mathcal{M}$ with the potential for stronger expressive abilities.*

**Assumption 3.3** *We consider practical scenarios that require high detectability, and thus a large value of $\delta$. In such a strong watermarking scenario, tokens from the green list are more probable to be used than those from the red list.*

**Note:** Assumption 3.3 establishes the foundational premise of text watermarking's effectiveness.

Building upon the Definitions and Assumption, we derive the following theorem.

**Theorem 3.4** *Consider that $\boldsymbol{p}_t \in \mathbb{R}^{|\mathcal{V}|}$ represents the predicted distribution of the model $\mathcal{M}$ at decoding time $t$. Let $w_i$ denote the token with the $i^{th}$*

*highest probability in $\mathbf{p}_t$. The higher the rank of a token (i.e., the smaller $i$ is), the more suitable it is to be selected. Under the conditions of Assumption 3.3, the WatME watermarking method is more likely to select a suitable token compared to the vanilla watermarking method.*

**Theorem 3.5** *Given a fixed proportion $\gamma$ of the green team, the expressive power of a language model $\mathcal{M}$ employing the WatME exceeds that of one utilizing a vanilla watermarking approach.*

These theorems highlight two advantages of WatME; their proofs are in the Appendix C.

## 4 Impact on Emergent Abilities

The majority of research on text watermarking utilizes the C4 dataset (Dodge et al., 2021) as a basis for testing perplexity (PPL). However, watermarking not only impacts the fluency of text generation but also holds the potential to influence LLMs on a broader scale, such as emergent abilities. These unique abilities intrinsic to LLMs garner significant interest from users and stimulate curiosity within the research community. However, they are often overlooked in the field of text watermarking.

Although a consensus definition is lacking, emergent abilities are typically characterized in many studies (Brown et al., 2020b; Wei et al., 2022; Yu et al., 2023) as a model's capacity to perform specific tasks without training. In light of this, we propose to test and compare the performance of WatME and Vanilla watermark algorithms on different tasks using prompting technologies.

To comprehensively test the impact of watermarking on these abilities, we attempt to categorize it into different scenarios for a more exhaustive examination. Specifically, we draw upon Cattell's cognitive theory (Cattell, 1963), which bifurcates intelligence into crystallized and fluid intelligence. Crystallized intelligence corresponds to the model's utilization of learned knowledge and experience, while fluid intelligence involves logical thinking and solving problems. Correspondingly, we propose to examine crystallized intelligence through an assessment of the model's knowledge capabilities, and fluid intelligence through its ability to reason and solve mathematical problems.

**Knowledge Capability.** To evaluate the model's mastery of world knowledge, we employ TruthfulQA (Lin et al., 2022), a benchmark designed to test if LLMs can generate truthful and informative answers. We select the generation setting.

**Reasoning Capability.** We employ the GSM8K dataset to assess the model's chain-of-thought reasoning. Comprising 8K arithmetic and math problems, it provides a platform for evaluating reasoning performance. Aligned with the CoT Hub prompt (Fu et al., 2023), our evaluations include few-shot scenarios that prompt the model to demonstrate reasoning and generate thought chains.

## 5 Experiments

### 5.1 Experimental Setups

**Evaluation Metrics** To evaluate detection performance, following previous work, we use the Area Under the Receiver Operating Characteristic curve (*AUROC*), a well-established metric for binary classifiers. For mathematical reasoning tasks, we use *Accuracy* to assess the correctness of the model's solutions. In our evaluation of the TruthfulQA dataset, following Lin et al. (2022), we use the trained GPT-Truth and GPT-Info scorers, assessing the model's capacity to generate both truthful and informative responses. Given the potential tradeoff between these two perspectives, the product of Truth and Information (*Truth.*Info.*) is commonly used as an overall measure of performance. On the C4 dataset, we report Perplexity (PPL).

**Baselines** We compared our model with four established baselines. First, KGW-Mark (Vanilla watermarking) (Kirchenbauer et al., 2023), which categorizes teams into 'red' and 'green' to facilitate detection. Second, Gumbel-Mark (Kuditipudi et al., 2023), which employs a Gumbel-Softmax distribution to introduce stochasticity into the watermarking process. Third, Unbiased-Mark (Hu et al., 2024), which implements reweighting techniques to maintain the expected output distribution of the LLM during watermarking. Lastly, Provable-Mark (Zhao et al., 2023), which uses a fixed hash key during watermarking to achieve provably better performance.

**Models** We utilized two distinct types of LLMs for experimentation: the non-aligned Llama2 model (Touvron et al., 2023), and the aligned Vicuna v1.5 model (Chiang et al., 2023). The majority of the results reported in this paper were obtained using the 7B version of the models.

Further setup details are in Appendix E.

| Model | GSM8K | | TruthfulQA | | | | C4 | |
|---|---|---|---|---|---|---|---|---|
| | Acc. | AUROC | True. | Info. | True.*Info. | AUROC | PPL | AUROC |
| Llama2-7B | 11.22 | - | 95.10 | 92.78 | 88.23 | - | 4.77 | - |
| + KGW-Mark | $5.61_{-50.0\%}$ | 0.8886 | $57.16_{-39.9\%}$ | $84.33_{-9.1\%}$ | $48.20_{-45.4\%}$ | 0.8416 | 7.00 | 0.9724 |
| + Gumbel-Mark | $7.28_{-35.1\%}$ | 0.9121 | $45.90_{-51.7\%}$ | $92.78_{-0.0\%}$ | $42.59_{-51.7\%}$ | 0.4931 | 39.93 | 0.9422 |
| + Unbiased-Mark | $10.24_{-8.7\%}$ | 0.5478 | $44.06_{-53.7\%}$ | $93.76_{+1.1\%}$ | $41.43_{-53.0\%}$ | 0.5051 | 15.62 | 0.5451 |
| + Provable-Mark | $5.16_{-54.01\%}$ | 0.9052 | $64.14_{-32.6\%}$ | $91.68_{-1.2\%}$ | $58.80_{-33.4\%}$ | 0.9555 | 10.21 | 0.9623 |
| + WatME$_{dictionary}$ | $9.17_{-18.3\%}$ | 0.8995 | $69.28_{-27.2\%}$ | $88.25_{-4.9\%}$ | $61.14_{-30.7\%}$ | 0.8848 | 5.32 | 0.9804 |
| + WatME$_{prompting}$ | $5.84_{-48.0\%}$ | 0.9128 | $55.83_{-41.3\%}$ | $95.10_{+2.5\%}$ | $50.39_{-42.9\%}$ | 0.8659 | 6.89 | 0.9724 |
| Vicuna-v1.5-7B | 17.51 | - | 93.88 | 87.27 | 81.92 | - | 10.77 | - |
| + KGW-Mark | $13.87_{-20.8\%}$ | 0.7870 | $74.05_{-21.1\%}$ | $87.52_{+0.3\%}$ | $64.81_{-20.1\%}$ | 0.7417 | 11.62 | 0.9679 |
| + Gumbel-Mark | $9.02_{-48.5\%}$ | 0.7077 | $68.30_{-27.2\%}$ | $87.27_{-0.0\%}$ | $59.61_{-27.2\%}$ | 0.4647 | 48.93 | 0.8617 |
| + Unbiased-Mark | $17.89_{+2.2\%}$ | 0.5508 | $70.38_{-25.0\%}$ | $88.86_{+1.8\%}$ | $62.54_{-23.7\%}$ | 0.4855 | 19.93 | 0.5000 |
| + Provable-Mark | $12.21_{-30.27\%}$ | 0.8020 | $74.42_{-20.7\%}$ | $96.70_{+10.8\%}$ | $71.96_{-12.2\%}$ | 0.8796 | 10.21 | 0.9582 |
| + WatME$_{dictionary}$ | $14.78_{-15.6\%}$ | 0.8044 | $78.95_{-15.9\%}$ | $97.43_{+11.6\%}$ | $76.92_{-6.1\%}$ | 0.7897 | 10.96 | 0.9582 |
| + WatME$_{prompting}$ | $16.22_{-7.4\%}$ | 0.7843 | $69.65_{-25.8\%}$ | $97.45_{-11.5\%}$ | $67.87_{-17.2\%}$ | 0.7396 | 11.54 | 0.9519 |

Table 1: Performance comparison of Llama2-7B and Vicuna-v1.5-7B under different watermarking algorithms.

### 5.2 Main Results

**Greater Impact on Emergent Abilities than Fluency** The experimental evidence suggests that watermarking notably hinders the emergent abilities of LLMs much more than fluency (see Table 1). Specifically, the non-aligned Llama2 model experienced a decline in performance exceeding 50% on both the GSM8K and TruthfulQA benchmarks. In contrast, the aligned model, Vicuna, demonstrated relative resilience but still incurred performance reductions greater than 20% on these benchmarks. This demonstrates the adverse impact of Vanilla Watermarking on the knowledge and reasoning capabilities of LLMs, with reasoning showing a marginally greater susceptibility.

**Superiority of WatME over baselines in Preserving Emergent Abilities** Across all models and benchmarks, the WatME consistently outperformed baseline watermarking methods. For the Llama2 model, WatME mitigated performance degradation by 16.8% on GSM8K and by 14.7% on TruthfulQA compared to the strongest baseline. Similarly, for the Vicuna model, the reductions were 13.4% and 14.0%, respectively. These outcomes underscore WatME's significant effectiveness in preserving the emergent capabilities of LLMs without compromising performance as significantly as other methods.

**Comparable Detection Performance of WatME** Despite the trade-off between text quality and detection performance, WatME's detection efficacy matched that of the Vanilla Watermark while also enhancing model capabilities, as evidenced by similar AUROC scores—suggesting our algorithm attained a better equilibrium than the baseline. In contrast, the Gumbel-Mark method noticeably compromised detection performance, particularly in aligned models and when generating short responses (TruthfulQA). Additionally, more performance results under different watermark strengths are presented in Discussion 6.3.

**Distinct Advantages of WatME Variations** It is evident that different WatME variations exhibit unique strengths; The 'dictionary' variant outperformed in the *Accuracy* and *Truthfulness* scores, while the 'prompting' variant excelled in the *Informativeness*. The integration of these variants may offer a fruitful avenue for future research. For a comprehensive understanding, a manual analysis of lexical clusters derived from these methods is presented in the Discussion 6.1.

**Alignment Diminishes Watermark Effectiveness** Surprisingly, aligned models showed significantly greater resistance to watermarking effects than non-aligned models. Specifically, Vicuna 1.5's performance dipped 30% less than Llama2's across all benchmarks, coupled with a 10% lower watermark detection performance. To understand the underlying reasons for these differences, we analyzed the output distribution discrepancies between aligned and unaligned models in the Discussion 6.4.

## 6 Discussion

### 6.1 Analysis of Clustering Methods

To analyse redundant clusters from diverse methods, we set evaluation criteria to ensure analytical

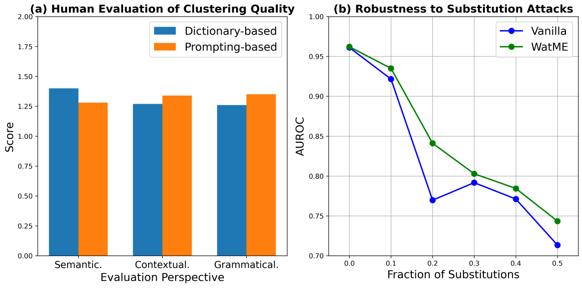


Figure 2: (a) Human evaluation for the quality of clusters mined by varied methods and (b) testing detection robustness against substitution attacks.

rigour. These criteria spanned *semantic consistency*, *contextual appropriateness*, and *grammatical consistency*, which are essential aspects of cluster quality. Two annotators used a rating scale of 0, 1, 2 to annotate the clusters. A score of '2' indicated high or ideal consistency, '1' denoted moderate or usable consistency, and '0' identified low or unusable consistency within a cluster. The kappa value for the annotations is 0.657. Figure 2(a) shows both methods met usability, but fell short of ideal effectiveness. The dictionary approach was superior in semantic coherence due to its utilization of lexical databases. Conversely, the prompting method outperformed in contextual and grammatical consistency, reflecting the varied linguistic corpus training of LLMs. This suggests the potential benefits of a combined approach, a topic reserved for future research.

### 6.2 Robustness Against Attacks

In addition to affecting the performance of LLMs, watermarks are also vulnerable to attacks aimed at their removal. To evaluate the robustness of our method, we conducted tests against two prevalent types of attacks: substitution attacks and paraphrase attacks. For the substitution attack, we evaluated 200 examples from GSM8k, with various token replacement ratios. As shown in Figure 2(b), WatME consistently outperformed the baseline method in the robustness of detection across different levels of token replacement. For paraphrase attacks, we use a powerful paraphraser, llama-2-chat-13B, to extensively rewrite the watermarked text generated by llama-2-7b. We provided it with the prompt: "Please paraphrase the following text, altering the wording significantly yet preserving the original meaning and length." We then subjected our system to these rewritten samples using 200 entries from both GSM8k and TruthfulQA. The results are presented in Tables 2.

| Method | Dataset | Original | Para. Attack |
|---|---|---|---|
| KGW-Mark | GSM8k | 0.885 | 0.745 |
| WatME | | 0.955 | 0.910 |
| KGW-Mark | TruthfulQA | 0.924 | 0.528 |
| WatME | | 0.949 | 0.673 |

Table 2: Detection robustness against paraphrasing attacks.

We offer two perspectives to understand the robustness of WatME: (1) Intuitively, for substitution attacks, the effect on watermarking depends on whether it triggers a token swap between the 'red' and 'green' teams: a swap affects the detection, while no swap means the watermark remains intact. With KGW-Mark, semantically similar tokens may be allocated to one team, resulting in a substitution invariably causing a swap. In contrast, WatME is intentionally designed to prevent this scenario. Therefore, the likelihood of a red-green swap—and consequently the impact on the watermark—is reduced in WatME compared to KGW. (2) From an encryption viewpoint, whereas KGW-Mark relies on a single division between teams, WatME employs multiple divisions—the number of clusters plus one (|C|+1), as outlined in Algorithm 2. Though these multiple partitions are computationally equivalent to a single partition due to efficient parallel matrix operations (explained in Appendix D), they introduce a higher level of complexity and robustness to the encryption process.

### 6.3 Performance Trade-offs at different Delta

The efficacy of the Watermark is influenced by the hyperparameter, Delta, which controls the watermark strength. An increase in Delta facilitates easier watermark detection but at the cost of severer impact on the LLMs. We analyse the TruthfulQA and GSM8K datasets. Figure 3 shows WatME consistently achieved a more favourable balance between watermark robustness and LLM performance across various Delta settings, surpassing Vanilla Watermark. Notably, the performance curves of WatME are strictly better than that of Vanilla, indicating that at equivalent watermark strengths, WatME always maintains superior performance compared to Vanilla Watermark.

### 6.4 Aligned vs Unaligned Models

Our examination of the response sensitivity to watermarking in aligned and unaligned models in-

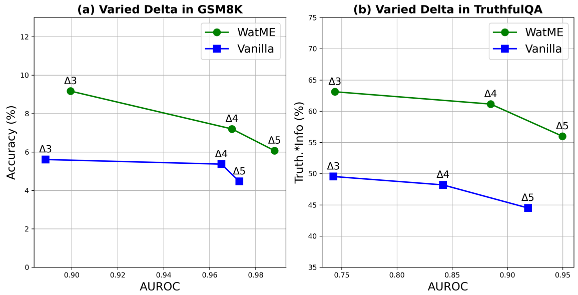


Figure 3: Performance trade-offs comparison between WatME and Vanilla Watermark on TruthfulQA and GSM8K at different Delta (Δ) values.

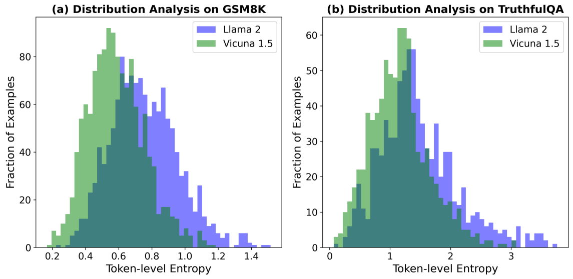


Figure 4: Token-level entropy distributions for aligned (green) and unaligned (blue) models on GSM8K and TruthfulQA benchmarks.

volved analyzing their output distributions on the TruthfulQA and GSM8K datasets. We computed the average entropy for tokens in the generated answers and found that aligned models exhibit markedly lower entropy, suggesting more deterministic response patterns, as illustrated in Figure 4. This pronounced certainty in aligned models' outputs presents a challenge for watermarking because of the limited variability that is essential for effective watermark encoding.

## 7 Conclusion

This study explores the impact of watermarking on the emergent abilities of LLMs—an aspect often neglected in the field. Our findings highlight the considerable adverse effects of traditional watermarking methods on LLMs' emergent abilities, including knowledge recall and logical reasoning.

In response, we introduced WatME—a novel watermarking approach that leverages lexical redundancy. Theoretical analysis and comprehensive empirical results indicate WatME consistently preserves the expressive power of LLMs without compromising detection performance, enabling developers to encode watermarks with less disruption to user experience.

The advancements with WatME mark a stride in lossless watermarking, enabling developers to encode watermarks with less disruption to user experience. We hope to promote a better equilibrium between regulatory compliance and user satisfaction in LLM development.

## Limitations

In this section, we discuss the limitations of this work from two perspectives.

Firstly, although WatME represents a step toward lossless watermarking, it is not entirely lossfree. The introduction of a controlled bias, inherent to watermarking methods, subtly alters the generated data. This compromise is a critical consequence as it diverges from the ideal of a completely lossless system. This deviation poses a dilemma for developers weighing the benefits of watermarking against potential user experience and regulatory trade-offs. Future work aims to bridge this gap, enhancing the WatME method to maintain output integrity and broaden its appeal for practical implementation.

Secondly, while our method is designed to be language-agnostic, the empirical validation presented in this work is limited to models processing the English language. We acknowledge that the applicability of watermarking across various linguistic contexts is critically important. Future investigations will endeavour to broaden the scope to include more languages, ensuring the generalizability and effectiveness of our approach in a multilingual context.

Thirdly, the challenge of watermarking in low-entropy scenarios remains an open problem. Our dataset encompasses a range of scenarios, including low-entropy situations where outcomes are more predictable and our methodology remains effective. However, embedding watermarks in text with universally recognized, low-entropy answers poses significant challenges, highlighting the need for further investigation into constructing and testing methodologies for low-entropy corpora.

Lastly, our LLMs-based cluster generation approach is influenced by the robustness of the prompting methods. Different prompt constructions can lead to varying outcomes (Zhao et al., 2021; Chen et al., 2023b, 2024), represents a limitation that warrants further discussion and exploration in future work.

Despite these limitations, we believe our work serves as a significant catalyst for the field, contributing positively to the advancement of more lossless and detectable text watermarking techniques.

## References

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. 2018. Turning your weakness into a strength: Watermarking deep neural networks by backdooring.

Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. 2001. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings 4*, pages 185–200. Springer.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Raymond B. Cattell. 1963. Theory of fluid and crystallized intelligence: A critical experiment. *Journal of Educational Psychology*, 54(1):1–22. ShortDOI: 10/fs6ptd KerkoCite.ItemAlsoKnownAs: 10.1037/h0046743 10/fs6ptd 1963-07991-001 2339240:TGQK3VJY 2405685:C8ZBFK3U.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023. On the possibilities of ai-generated text detection.

Liang Chen, Yatao Bian, Li Shen, and Kam-Fai Wong. 2024. Simple permutations can fool LLaMA: Permutation attack and defense for large language models. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Liang Chen, Yang Deng, Yatao Bian, Zeyu Qin, Bingzhe Wu, Tat-Seng Chua, and Kam-Fai Wong. 2023a. Beyond factuality: A comprehensive evaluation of large language models as knowledge generators. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6325–6341, Singapore. Association for Computational Linguistics.

Liang Chen, Hongru Wang, Yang Deng, Wai Chung Kwan, Zezhong Wang, and Kam-Fai Wong. 2023b. Towards robust personalized dialogue generation via order-insensitive representation regularization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7337–7345, Toronto, Canada. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable watermarks for language models. *arXiv preprint arXiv:2306.09194*.

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. 2023. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10602–10621, Singapore. Association for Computational Linguistics.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2022. Is gpt-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. 2023. Three bricks to consolidate watermarks for large language models.

Yao Fu, Litu Ou, Mingyu Chen, Yuhao Wan, Hao Peng, and Tushar Khot. 2023. Chain-of-thought hub: A continuous effort to measure large language models' reasoning performance. *CoRR*, abs/2305.17306.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. Gltr: Statistical detection and visualization of generated text. In *Annual Meeting of the Association for Computational Linguistics*.

Philipp Hacker, Andreas Engel, and Marco Mauer. 2023. Regulating chatgpt and other large generative AI models. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency,* *FAccT 2023, Chicago, IL, USA, June 12-15, 2023*, pages 1112–1123. ACM.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. 2022. Protecting intellectual property of language generation apis with lexical watermark. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10758–10766.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2024. Unbiased watermark for large language models. In *The Twelfth International Conference on Learning Representations*.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. In *International Conference on Computational Linguistics*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. *International Conference on Machine Learning*.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust distortion-free watermarks for language models.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Shuaiyi Li, Yang Deng, Deng Cai, Hongyuan Lu, Liang Chen, and Wai Lam. 2024. Consecutive model editing with batch alongside hook layers.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Y. Zou. 2023. Gpt detectors are biased against non-native english writers. *ArXiv*, abs/2304.02819.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *ArXiv*, abs/2301.11305.

N. Nikolaidis and I. Pitas. 1999. Digital image watermarking: an overview. In *Proceedings IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 1–6 vol.1.

OpenAI. 2019. Gpt-2: 1.5b release.

OpenAI. 2023a. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

OpenAI. 2023b. New ai classifier for indicating ai-written text. *OpenAI blog*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. 2018. Deepsigns: A generic watermarking framework for ip protection of deep learning models.

Vinu Sankar Sadasivan, Aounon Kumar, S. Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *ArXiv*, abs/2303.11156.

S. Samuel and W.T. Penzhorn. 2004. Digital watermarking for copyright protection. In *2004 IEEE Africon. 7th Africon Conference in Africa (IEEE Cat. No.04CH37590)*, volume 2, pages 953–957 Vol.2.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units.

Katzenbeisser Stefan, AP Fabien, et al. 2000. Information hiding techniques for steganography and digital watermarking.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Honai Ueoka, Yugo Murawaki, and Sadao Kurohashi. 2021. Frustratingly easy edit-based linguistic steganography with a masked language model. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5486–5492, Online. Association for Computational Linguistics.

Hongru Wang, Lingzhi Wang, Yiming Du, Liang Chen, Jingyan Zhou, Yufei Wang, and Kam-Fai Wong. 2023. A survey of the evolution of language model-based dialogue systems.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Alex Wilson, Phil Blunsom, and Andrew D Ker. 2014. Linguistic steganography on twitter: hierarchical language modeling with manual interaction. In *Media Watermarking, Security, and Forensics 2014*, volume 9028, pages 9–25.

Max Wolff. 2020. Attacking neural text detectors. *ArXiv*, abs/2002.11768.

Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. 2022. Tracing text provenance via context-aware lexical substitution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11613–11621.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2021. Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5482–5492, Online. Association for Computational Linguistics.

# Appendix

## A Algorithms of Watermark

This section presents detailed algorithms for the watermark encoding and detection processes as outlined in (Kirchenbauer et al., 2023). Algorithm 2 delineates the procedure for encoding a watermark into the output sequence generated by a language model. Conversely, Algorithm 3 explicates the method for detecting and confirming the watermark's presence within generated sequences.

**Algorithm 2** Vanilla Watermark Encoding

```
Input: prompt $x_1 \cdots x_m$,
       green list size $\gamma \in (0, 1)$,
       watermark strength $\delta > 0$.
for $t = 0, 1, \cdots, T - 1$ do
    1. Get the logit $\boldsymbol{\ell}_t \in \mathbb{R}^{|\mathcal{V}|}$ from $\mathcal{M}$.
    2. Use the hash of the previous token as the
       random seed to partition the vocabulary
       of $\mathcal{M}$ into a "green list" $G_t$ of size $\gamma|\mathcal{V}|$,
       and a "red list" $R_t$ of size $(1-\gamma)|\mathcal{V}|$.
    3. Add $\delta$ to each green list logit and then
       apply softmax to the modified logits.
           $\hat{\boldsymbol{\ell}}_t[i] := \boldsymbol{\ell}_t[i] + \delta, i \in G_t$
           $\hat{\boldsymbol{p}}_t = softmax(\hat{\boldsymbol{\ell}}_t)$
    4. Sample a next token $y_{t+1}$ from $\hat{\boldsymbol{p}}_t$.
end for
Output: watermarked text $y_1 \cdots y_T$.
```

## B Prompt for Cluster Mining

To facilitate the generation of synonym clusters, we employed Llama2-13B-chat. The approach involved crafting a prompt (Figure 5) that combines a clear task description with a set of demonstrations designed to illustrate the desired task. By presenting the model with a few-shot example, we primed Llama2-13B-chat to understand and perform the specific task of synonym generation. The few-shot prompt was crucial for the model to recognize the pattern and replicate it for new target

**Algorithm 3** Vanilla Watermark Detection

```
Input: text $\boldsymbol{y}$, detection threshold $\tau$.
1. Use the previous token to find the "green list" $G_t$ at the step $t$ as in Alg. 2.
2. Calculate the number of green tokens in $\boldsymbol{y}$ as $|\boldsymbol{y}|_G = \sum_{t=1}^{n} \mathbb{1}(y_t \in G)$.
3. Compute the $z$-statistic:
       $z_{\boldsymbol{y}} = (|\boldsymbol{y}|_G - \gamma|\mathcal{V}|) \,/\, \sqrt{|\mathcal{V}|\gamma(1-\gamma)}$.
4. if $z_{\boldsymbol{y}} > \tau$ then return 1 (watermarked).
5. else return 0 (unwatermarked).
Output: 0 or 1
```

words, thus enabling the mining of synonym clusters effectively.

## C Proofs of Theorems

In this section, we present the detailed proofs of the theorems introduced before. Each theorem is treated in its respective subsection.

### C.1 Proof of Theorem 3.4

**Proof** We begin the proof by considering $i = 1, 2$.

**Case I: where $w_1$ is in the green list ($G_t$):**

If $w_1 \in G_t$, then both watermarking methods are lossless because they can select the most suitable token $w_1$.

**Case II: where $w_1$ is in the red list ($R_t$):**

We consider $w_2$, which may or may not be a synonym of $w_1$:

**Sub-case i:** $w_2$ is not a synonym of $w_1$.

If $w_1 \notin G_t$ and $\nexists\, C_i \in \mathcal{C}$ s.t. $w_1, w_2 \in C_i$, then according to Algo. 1 we have:

$$P_{WatME}(w_2 \in G_t) = P_{watermark}(w_2 \in G_t).$$

In this case, the two methods are the same.

**Sub-case ii:** $w_2$ is a synonym of $w_1$.

If $w_1 \notin G_t$ and $\exists C_i \in \mathcal{C}$ s.t. $w_1, w_2 \in C_i$, then according to Algo. 1 we have:

$$P_{WatME}(w_2 \in G_t) > P_{watermark}(w_2 \in G_t).$$

Based on Assumption 3.3, WatME is more likely to select the suitable token. Combining these cases, the theorem is proven. It should be noted that while this proof explicitly considers the cases for $i = 1, 2$, the logic extends to any arbitrary value of $i$.

### C.2 Proof of Theorem 3.5

**Proof** Let us define the vocabulary $V$ with synonym clusters $S = \{C_1, \ldots, C_n\}$, where $\bar{C}$ represents the set of non-synonymous, unique words. According to Algs 2 and 1, WatME maintains a constant number of distinct semantic representations, quantified as $n + \gamma \cdot |\bar{C}|$. In contrast, the semantic token count of standard watermarking algorithms is lower than this figure. According to Definition 3.1 the disparity in semantic entropy between the two methodologies is thus evident. Given Definition 3.2, the increased semantic entropy inherent to WatME confirms the theorem.

## D Time Complexity Analysis

The conventional algorithm necessitates a partition of the vocabulary per decoding operation, which results in a time complexity of $O(|V|)$. Our method incorporates two partitioning stages: initially targeting the redundant cluster, followed by the remaining vocabulary. During the first stage, we pad the cluster into a 2D matrix and conduct parallel sampling. The subsequent stage aligns with the procedures of the Vanilla algorithm. Consequently, the time complexity of our method remains at $O(|V|)$.

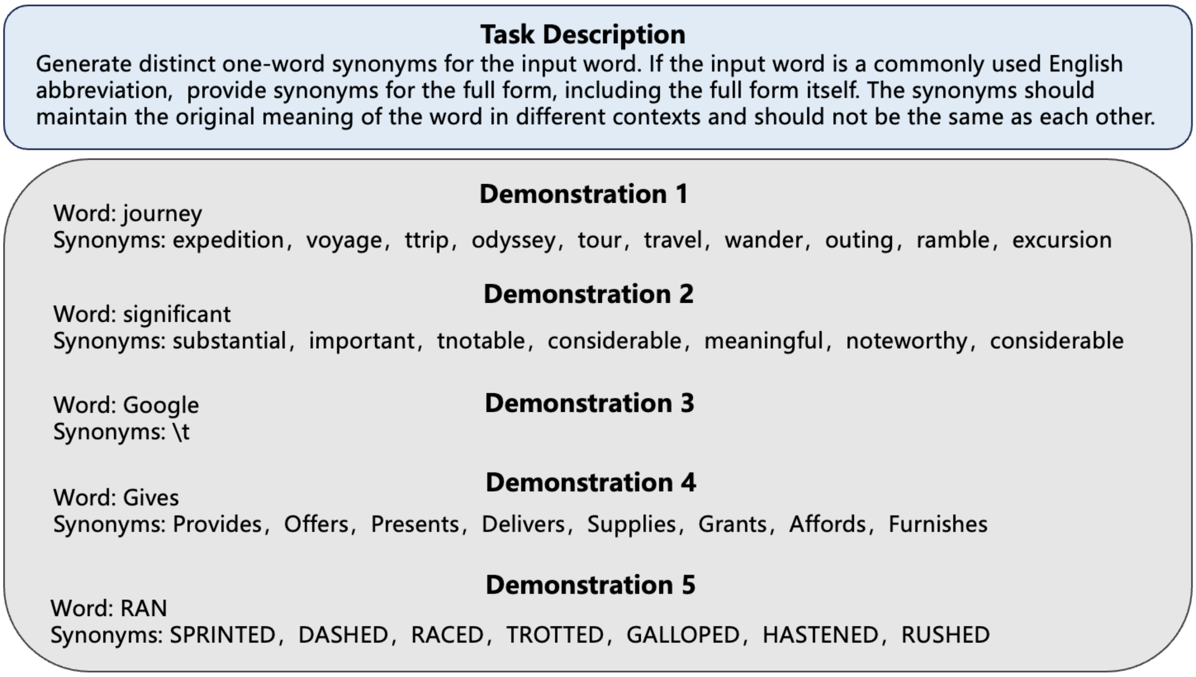
## E Setup Details

In our experiments, we used prompts from the CoT hub (Fu et al., 2023) for the GSM8K dataset and the original prompts from TruthfulQA (Lin et al., 2022). The Llama2 model was evaluated using its original prompt format to maintain consistency. Greedy decoding was employed as the strategy for all tasks, with maximum decoding lengths set at 128 tokens for GSM8K and 50 tokens for TruthfulQA, which allowed for the complete generation of answers within the datasets.

To account for the differing answer lengths in the GSM8K and TruthfulQA datasets, we fine-tuned the watermark hyperparameters. For GSM8K, with its longer answers aiding detection, we used a milder watermark intensity, setting gamma at 0.3 and delta at 3.0. Conversely, the brevity of answers in TruthfulQA complicates detection, necessitating a stronger watermark intensity—again with gamma at 0.3, but with delta increased to 4.0 to achieve satisfactory detection performance (AUROC > 0.7).

Evaluation metrics were carefully chosen: AUROC was calculated using the 'sklearn' library, and for the assessment of GPT-Truth and GPT-Info, we utilized a fine-tuned Llama2-13B-chat model that demonstrated an accuracy above 93% on the validation set. All model implementations were executed using the 'transformers' library.

The hardware employed for these experiments



**Task Description**

Generate distinct one-word synonyms for the input word. If the input word is a commonly used English abbreviation, provide synonyms for the full form, including the full form itself. The synonyms should maintain the original meaning of the word in different contexts and should not be the same as each other.

**Demonstration 1**

Word: journey
Synonyms: expedition，voyage，ttrip，odyssey，tour，travel，wander，outing，ramble，excursion

**Demonstration 2**

Word: significant
Synonyms: substantial，important，tnotable，considerable，meaningful，noteworthy，considerable

**Demonstration 3**

Word: Google
Synonyms: \t

**Demonstration 4**

Word: Gives
Synonyms: Provides，Offers，Presents，Delivers，Supplies，Grants，Affords，Furnishes

**Demonstration 5**

Word: RAN
Synonyms: SPRINTED，DASHED，RACED，TROTTED，GALLOPED，HASTENED，RUSHED

Figure 5: Few-Shot Demonstration of Synonym Generation using LLMs.

consisted of a 40GB A100 GPU and a 32GB V100 GPU, ensuring sufficient computational power for model training and evaluation.

## F Examples of Redundant Clusters

We present some examples of mined clusters at 3.

## G Multilingual Performance Testing

We expand our evaluation to include the Chinese Long Text Summarization Dataset (CLTS) and a bilingual Large Language Model (LLM), ChatGLM3-6b. This model employs Byte Pair Encoding (BPE) tokenization with a vocabulary size of 65k—double that of the Llama 2 model which has a 32k vocabulary size. Synonym mining, a critical step in our process, was conducted using the ChatGLM3-13B model. The performance of different watermarking methods was evaluated using the ROUGE-L and AUROC metrics, as shown in Table 4. The results highlight that watermarking with WatME considerably enhances detection robustness compared to the baseline method, maintaining effectiveness despite varying levels of token replacement. This improvement underscores WatME's capability to integrate seamlessly without compromising the natural language generation quality.

| **Dictionary-based Method** | **LLM-based Method** |
|---|---|
| ’should’, ’must’, ’would’ | ’must’, ’ought’, ’should’ |
| ’job’, ’pursuit’, ’operation’, ’profession’, ’career’, ’employment’, ’behavior’ | ’job’, ’task’, ’work’ |
| ’inside’, ’in’ | ’_inside’, ’_inner’, ’_within’ |

Table 3: Examples of Redundant Clusters.

| Method | ROUGE-L | AUROC |
|---|---|---|
| ChatGLM 3-6b | 11.29 | - |
| +KGW-Mark | 8.89 | 0.8415 |
| +WatME_prompting | 10.23 | 0.8514 |

Table 4: CLTS Results