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 an important technique for detecting machine-generated text. However, existing methods generally use arbitrary vocabulary partitioning during decoding, which results in the absence of appropriate words during the response generation process and disrupts the language model's expressiveness, thus severely degrading the quality of text response. To address these issues, we introduce a novel approach, Watermarking with Mutual Exclusion (WatME). Specifically, by leveraging linguistic prior knowledge of inherent lexical redundancy, WatME can dynamically optimize the usage of available vocabulary during the decoding process of language models. It employs a mutually exclusive rule to manage this redundancy, avoiding situations where appropriate words are unavailable and maintaining the expressive power of large language models (LLMs). We present theoretical analysis and empirical evidence demonstrating that WatME substantially preserves the text generation ability of LLMs while maintaining watermark detectability. Specifically, we investigate watermarking's impact on the emergent abilities of LLMs, including knowledge recall and logical reasoning. Our comprehensive experiments confirm that WatME consistently outperforms existing methods in retaining these crucial capabilities of LLMs.

1 INTRODUCTION

The advent of LLMs (Ouyang et al., 2022; OpenAI, 2023) with human-level generative capabilities presents vast opportunities across diverse NLP tasks (Deng et al., 2023; Li et al., 2024; Lyu et al., 2024). However, these advancements also bring to light concerns over potential misuse, such as misinformation dissemination (Chen et al., 2023a), information redundancy (Li & Li, 2024) and facilitation of academic dishonesty (Stokel-Walker, 2022), highlighting the need for reliable techniques to attribute AI-generated text to its origins.

Current text watermarking algorithms, advocating direct intervention in the generative process to embed identifiable fingerprints in machine-generated text, provide provenance verification (Kirchenbauer et al., 2023; Christ et al., 2023; Zhao et al., 2023). While this approach proves more effective in detecting LLM-generated content (Sadasivan et al., 2023), it often compromises text quality, posing a significant challenge for developers - how to effectively watermark while preserving text quality.

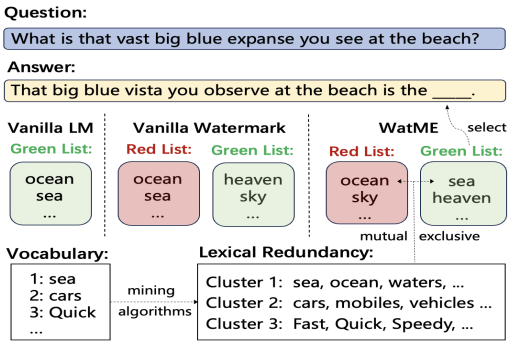

Figure 1: WatME's lossless watermarking benefit.

Recent efforts to enhance text quality in watermarking have focused on maintaining unbiased output distributions through pseudorandom perturbations or reweighting (Kuditipudi et al., 2023; Hu et al., 2024). These strategies, how-

ever, don't consistently assure superior text quality and may compromise the efficacy of watermark detection (Kuditipudi et al., 2023), especially in aligned models, thus reducing their applicability.

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 & Pitas, 1999; Samuel & 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 might 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 analysis(§ C) and empirical (§ 5) evidence supports their effectiveness without compromising detection capabilities. These improvements significantly bolster the emergent abilities of large models under watermarks, surpassing the performance of established baselines.

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). Nevertheless, existing detectors are susceptible to sophisticated adversarial strategies and tend to exhibit bias against individuals whose first language is not English (Wolff, 2020; 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 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 pseudorandomness-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

This section investigates lexical redundancy and its potential for improving watermark algorithms in language models. Preliminary and mathematical analyses are detailed in the Appendix A and C.

3.1 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, pre-constructed 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.

3.2 EXPLORE THE REDUNDANCY IN LEXICAL SPACE

Confronted with the unique challenge that text data's discrete and limited nature presents, unlike image data with its redundant pixels, we pivot to the concept of lexical redundancy. This involves tapping into the extensive vocabulary at a language model's disposal during the decoding process.

To utilize lexical redundancy for watermarking, we construct clusters of synonyms from the model's vocabulary through two primary methods: the dictionary-based approach, which draws upon authoritative lexicons such as WordNet, and the prompting-based approach, wherein advanced models like

LLaMA2 are prompted to provide context-aware synonyms. The complexities involved in forming these high-quality clusters are further explicated in Appendix B.1.

3.3 WatME: Exploit the Lexical Redundancy

Within the lexical space of a language model $\mathcal{M}$ with vocabulary $\mathcal{V}$, we define a subset $S \subseteq \mathcal{V}$ consisting of tokens that are synonymous. We denote a set of redundant lexical clusters as $C = \{C_i \mid i = 1..n\}$, where $\bigcup_{i=1}^{n} C_i = S$ and each $C_i = \{s_{ij} \mid j = 1..m_i, s_{ij} \in S\}$. Tokens $s_{ij}, s_{ik} \in C_i$ are considered interchangeable.

Using this redundancy, we introduce a mutual exclusion rule for watermarking: when tokens from cluster $\mathcal{A}$ are assigned to the red list, their synonyms $\mathcal{B}$ are assigned to the green list, and vice versa.

The WatME encoding algorithm, outlined in Alg. 1, uses a two-step process to generate green (G'_t, G_t) and red (R'_t) lists. Initial partitioning within clusters C assigns tokens to G'_t by γ. The next step divides the remaining vocabulary $\mathcal{V} \setminus S$ into G_t and R_t following γ. The ensuing steps mirror the standard watermarking as in original algorithm in appendix B. The WatME detection algorithm remains as detailed in B, with the green list now calculated per Alg. 1.

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**
  Get the logit $\boldsymbol{\ell}_t \in \mathbb{R}^{|\mathcal{V}|}$ from $\mathcal{M}$.
  Use seed from the last token, split each cluster C_i in parallel into a green list G'_{it} (of size $|C_i|\gamma$) and a red list R'_{it} (of size $(1-\gamma)|C_i|$).
  Let $G'_t = \cup_i G'_{it}$ and $R'_t = \cup_i R'_{it}$.
  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 $(1-\gamma)|V| - |R'_t|$.
  Merge lists from the previous two steps: $G_t = G_t \cup G'_t$ and $R_t = R_t \cup R'_t$.
  Add δ to the elements of logit $\boldsymbol{\ell}_t$ corresponding to the green list, then softmax.
  $\hat{\boldsymbol{p}}_t = \text{softmax}(\boldsymbol{\ell}_t[i] + \delta), i \in G_t$
  Sample the next token y_{t+1} from $\hat{\boldsymbol{p}}_t$.
end for
Output: watermarked text $y_1 \cdots y_T$.

4 Impact on Emergent Abilities

Research on text watermarking has primarily focused on fluency using datasets like C4 (Dodge et al., 2021), yet the impact on LLMs' emergent abilities—crucial to user engagement—has been largely ignored. Our study extends beyond C4, assessing how watermarking affects key cognitive functions as outlined by Cattell's theory (Cattell, 1963): crystallized intelligence, which pertains to knowledge application, and fluid intelligence, related to problem-solving and logic. We use TruthfulQA (Lin et al., 2022) to evaluate the model's ability to recall and provide accurate information. In parallel, the GSM8K (Cobbe et al., 2021) dataset serves to test the model's logical reasoning and problem-solving skills in few-shot learning scenarios.

5 Experiments

We validate our WatME's superiority across three scenarios and two distinct model types. Experimental setup details are in Appendix H, with further analysis in Appendix D.

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.

Table 1: Performance comparison of Llama2 and Vicuna v1.5 with various watermarking algorithms.

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	$5.61_{-50.0\%}$	0.889	$57.16_{-39.9\%}$	$84.33_{-9.1\%}$	$48.20_{-45.4\%}$	0.842	7.00	0.972
+ GUMBEL-MARK	$7.28_{-35.1\%}$	0.912	$45.90_{-51.7\%}$	$92.78_{-0.0\%}$	$42.59_{-51.7\%}$	0.493	39.93	0.942
+ UNBIASED-MARK	$10.24_{-8.7\%}$	0.548	$44.06_{-53.7\%}$	$93.76_{+1.1\%}$	$41.43_{-53.0\%}$	0.505	15.62	0.545
+ PROVABLE-MARK	$5.16_{-54.01\%}$	0.905	$64.14_{-32.6\%}$	$91.68_{-1.2\%}$	$58.80_{-33.4\%}$	0.956	10.21	0.962
+ WATME$_{dictionary}$	$9.17_{-18.3\%}$	0.900	$69.28_{-27.2\%}$	$88.25_{-4.9\%}$	$61.14_{-30.7\%}$	0.885	5.32	0.980
+ WATME$_{prompting}$	$5.84_{-48.0\%}$	0.913	$55.83_{-41.3\%}$	$95.10_{+2.5\%}$	$50.39_{-42.9\%}$	0.866	6.89	0.972
VICUNA-V1.5-7B	17.51	-	93.88	87.27	81.92	-	10.77	-
+ KGW-MARK	$13.87_{-20.8\%}$	0.787	$74.05_{-21.1\%}$	$87.52_{+0.3\%}$	$64.81_{-20.1\%}$	0.7417	11.62	0.968
+ GUMBEL-MARK	$9.02_{-48.5\%}$	0.708	$68.30_{-27.2\%}$	$87.27_{-0.0\%}$	$59.61_{-27.2\%}$	0.4647	48.93	0.862
+ UNBIASED-MARK	$17.89_{+2.2\%}$	0.551	$70.38_{-25.0\%}$	$88.86_{+1.8\%}$	$62.54_{-23.7\%}$	0.4855	19.93	0.500
+ PROVABLE-MARK	$12.21_{-30.27\%}$	0.802	$74.42_{-20.7\%}$	$96.70_{+10.8\%}$	$71.96_{-12.2\%}$	0.8796	10.21	0.958
+ WATME$_{dictionary}$	$14.78_{-15.6\%}$	0.804	$78.95_{-15.9\%}$	$97.43_{+11.6\%}$	$76.92_{-6.1\%}$	0.7897	10.96	0.958
+ WATME$_{prompting}$	$16.22_{-7.4\%}$	0.784	$69.65_{-25.8\%}$	$97.45_{-11.5\%}$	$67.87_{-17.2\%}$	0.7396	11.54	0.952

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 strength are presented in Appendix D.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 Appendix D.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 Appendix D.4.

6 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.

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 loss-free. 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), 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. Turning your weakness into a strength: Watermarking deep neural networks by backdooring, 2018.

Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. 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*, pp. 185–200. Springer, 2001.

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. Language models are few-shot learners, 2020.

Raymond B. Cattell. Theory of fluid and crystallized intelligence: A critical experiment. *Journal of Educational Psychology*, 54(1):1–22, February 1963. ISSN 0022-0663. doi: 10.1037/h0046743. URL https://ezp.lib.cam.ac.uk/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1963-07991-001&site=ehost-live&scope=site. short-DOI: 10/fs6ptd KerkoCite.ItemAlsoKnownAs: 10.1037/h0046743 10/fs6ptd 1963-07991-001 2339240:TGQK3VJY 2405685:C8ZBFK3U.

Liang Chen, Yang Deng, Yatao Bian, Zeyu Qin, Bingzhe Wu, Tat-Seng Chua, and Kam-Fai Wong. Beyond factuality: A comprehensive evaluation of large language models as knowledge generators. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6325–6341, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.390. URL https://aclanthology.org/2023.emnlp-main.390.

Liang Chen, Hongru Wang, Yang Deng, Wai Chung Kwan, Zezhong Wang, and Kam-Fai Wong. Towards robust personalized dialogue generation via order-insensitive representation regularization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 7337–7345, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.462. URL https://aclanthology.org/2023.findings-acl.462.

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. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10602–10621, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.711. URL https://aclanthology.org/2023.findings-emnlp.711.

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

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

Yao Fu, Litu Ou, Mingyu Chen, Yuhao Wan, Hao Peng, and Tushar Khot. Chain-of-thought hub: A continuous effort to measure large language models' reasoning performance. *CoRR*, abs/2305.17306, 2023. doi: 10.48550/arXiv.2305.17306. URL https://doi.org/10.48550/arXiv.2305.17306.

Philipp Hacker, Andreas Engel, and Marco Mauer. 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*, pp. 1112–1123. ACM, 2023. URL https://doi.org/10.1145/3593013.3594067.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. Protecting intellectual property of language generation apis with lexical watermark. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10758–10766, Jun. 2022. doi: 10.1609/aaai.v36i10.21321. URL https://ojs.aaai.org/index.php/AAAI/article/view/21321.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uWVC5FVidc.

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

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

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

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

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

Xianming Li and Jing Li. Generative deduplication for socia media data selection, 2024.

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

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

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

Yougang Lyu, Lingyong Yan, Shuaiqiang Wang, Haibo Shi, Dawei Yin, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. Knowtuning: Knowledge-aware fine-tuning for large language models, 2024.

George A. Miller. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992. URL https://aclanthology.org/H92-1116.

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

N. Nikolaidis and I. Pitas. Digital image watermarking: an overview. In *Proceedings IEEE International Conference on Multimedia Computing and Systems*, volume 1, pp. 1–6 vol.1, 1999. doi: 10.1109/MMCS.1999.779111.

OpenAI. Gpt-2: 1.5b release. November 2019. URL https://openai.com/research/gpt-2-1-5b-release.

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

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. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022.

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

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

S. Samuel and W.T. Penzhorn. Digital watermarking for copyright protection. In *2004 IEEE Africon. 7th Africon Conference in Africa (IEEE Cat. No.04CH37590)*, volume 2, pp. 953–957 Vol.2, 2004. doi: 10.1109/AFRICON.2004.1406827.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5149–5152, 2012. URL https://api.semanticscholar.org/CorpusID:22320655.

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

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

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

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. Llama 2: Open foundation and fine-tuned chat models, 2023.

Honai Ueoka, Yugo Murawaki, and Sadao Kurohashi. Frustratingly easy edit-based linguistic steganography with a masked language model. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5486–5492, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.433. URL https://aclanthology.org/2021.naacl-main.433.

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

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

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

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

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12697–12706. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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)*, pp. 5482–5492, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.426. URL https://aclanthology.org/2021.acl-long.426.

APPENDIX

A 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 B.

The watermark encoding process is guided by two parameters: γ and δ. 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 γ determines the size of the green list, while δ specifies the degree of encouragement for the green list, the increase in current logits $\boldsymbol{\ell}_t$ before performing softmax, as Eq.1. As δ 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 δ 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.

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

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

B 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 δ to each green list logit and then apply softmax to the modified logits.
$$\begin{aligned} \hat{\boldsymbol{\ell}}_t[i] &:= \boldsymbol{\ell}_t[i] + \delta, i \in G_t \\ \hat{\boldsymbol{p}}_t &= softmax(\hat{\boldsymbol{\ell}}_t) \end{aligned}$$
4. Sample a next token y_{t+1} from $\hat{\boldsymbol{p}}_t$.

end for
Output: watermarked text $y_1 \cdots y_T$.

Algorithm 3 Vanilla Watermark Detection

Input: text $\boldsymbol{y}$, detection threshold τ.

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:
$$z_{\boldsymbol{y}} = \left(|\boldsymbol{y}|_G - \gamma|\mathcal{V}|\right) / \sqrt{|\mathcal{V}|\gamma(1-\gamma)}.$$
4. **if** $z_{\boldsymbol{y}} > \tau$ **then return** 1 (watermarked).
5. **else return** 0 (unwatermarked).

Output: 0 or 1

B.1 Explore the Redundancy in Lexical Space

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., 2020), with the demonstrations being annotated manually by us. Detailed prompts are deferred to Appendix E.

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 & Nakajima, 2012; Kudo & 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 D.1.

C 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.

Definition C.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 C.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 C.3 *We consider practical scenarios that require high detectability, and thus a large value of δ. In such a strong watermarking scenario, tokens from the green list are more probable to be used than those from the red list.*

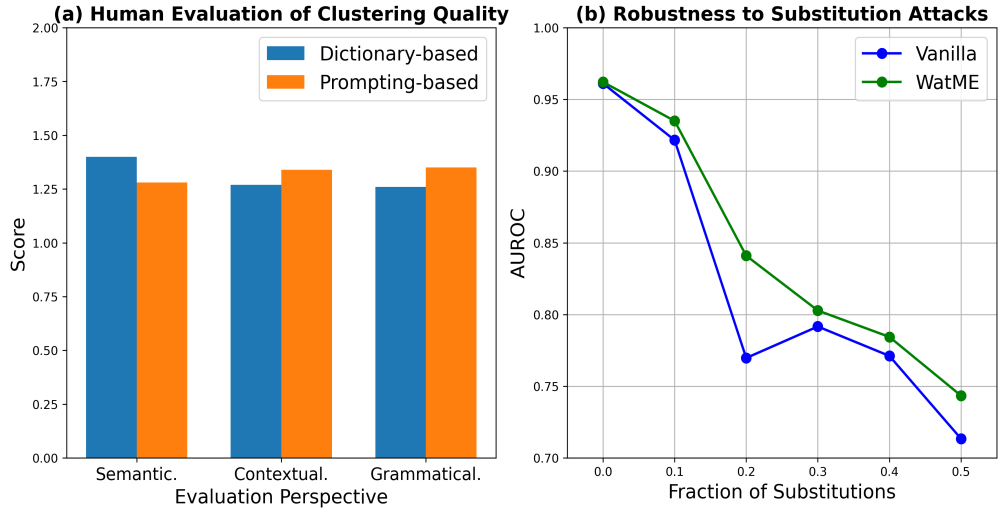

Figure 2: (a) Human evaluation for the quality of clusters mined by varied methods and (b) testing detection robustness against substitution attacks.

Note: Assumption C.3 establishes the foundational premise of text watermarking's effectiveness. Building upon the Definitions and Assumption, we derive the following theorem.

Theorem C.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 $\boldsymbol{p}_t$. 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 C.3, the WatME watermarking method is more likely to select a suitable token compared to the vanilla watermarking method.*

Theorem C.5 *Given a fixed proportion γ 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 F.

D Discussion

D.1 Analysis of Clustering Methods

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

D.2 Robustness Against Attacks

Within the scope of watermark robustness against common rewriting attacks, our study evaluated the resilience of the proposed WatME method compared to baseline watermarking techniques. In a simulated black-box attack scenario, where attackers were blind to the watermark encryption algorithm, we assessed the integrity of watermarks after random substitutions of text tokens. Utilizing a sample

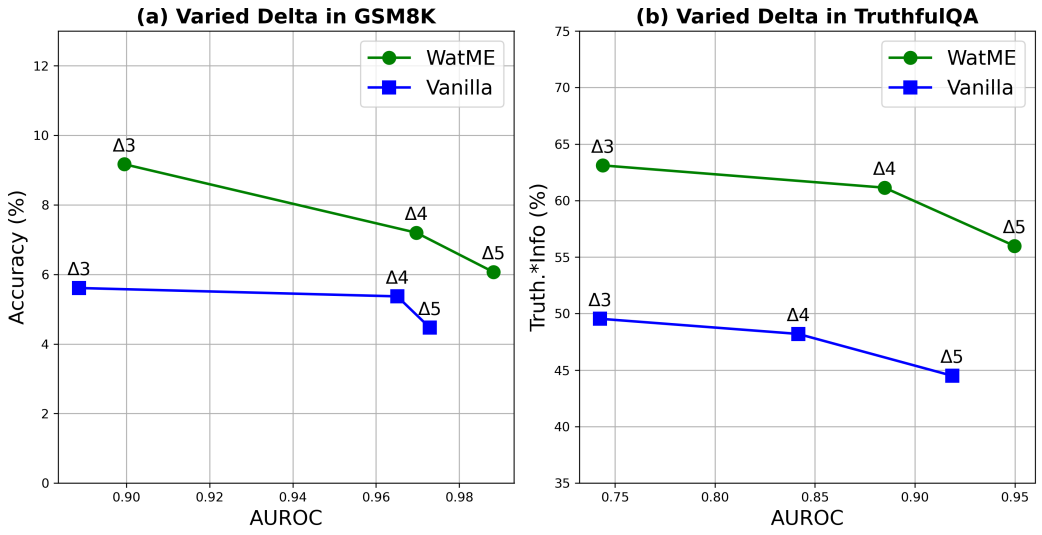

Figure 3: Performance trade-offs comparison between WatME and Vanilla Watermark on TruthfulQA and GSM8K at different Delta (Δ) values.

of 200 examples from the GSM8k dataset, the analysis, illustrated in Figure 2(b), demonstrated that WatME consistently outperformed the vanilla method in detection robustness across a spectrum of replacement ratios.

D.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 severe 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.

D.4 Aligned vs Unaligned Models

Our examination of the response sensitivity to watermarking in aligned and unaligned models involved analyzing their output distributions on the TruthfulQA and GSM8K datasets. We computed the average entropy for token 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.

E 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 words, thus enabling the mining of synonym clusters effectively.

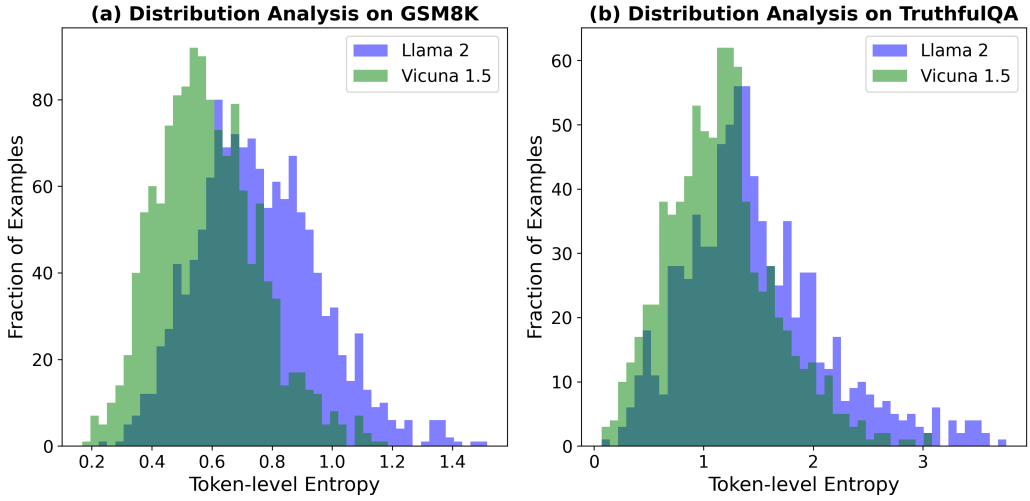

Figure 4: Token-level entropy distributions for aligned (green) and unaligned (blue) models on GSM8K and TruthfulQA benchmarks.

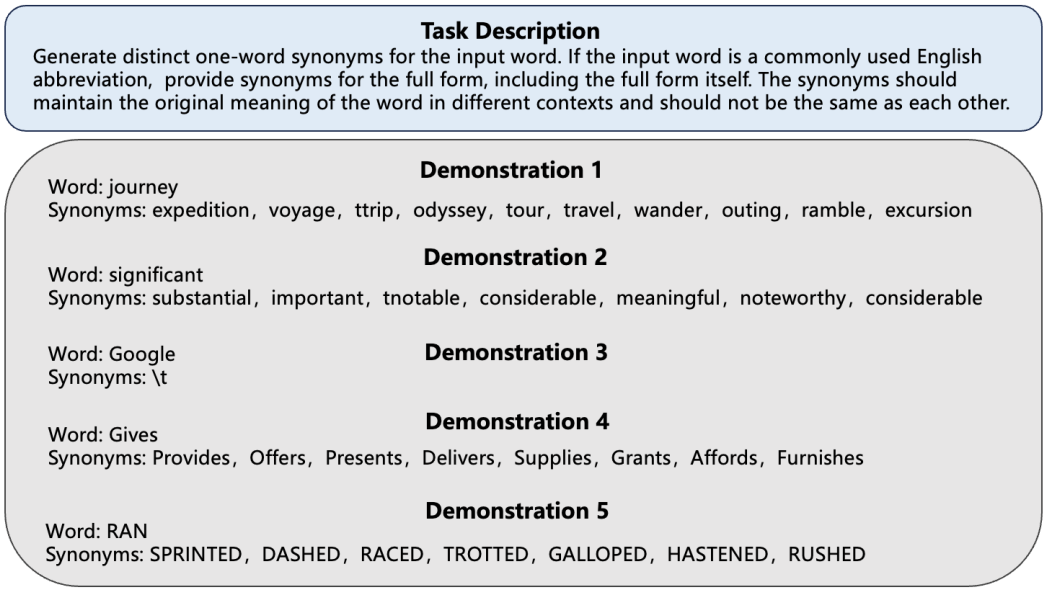

Figure 5: Few-Shot Demonstration of Synonym Generation using LLMs.

F Proofs of Theorems

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

F.1 Proof of Theorem C.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 C.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.

F.2 Proof of Theorem C.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 C.1 the disparity in semantic entropy between the two methodologies is thus evident. Given Definition C.2, the increased semantic entropy inherent to WatME confirms the theorem.

G 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 $O(|V|)$.

H Setup Details

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 trade-off 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 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.

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'

Figure 6: Examples of Redundant Clusters.

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

I Examples of Redundant Clusters

We present some examples of mined clusters at 6.