GumbelSoft: Diversified Language Model Watermarking via the GumbelMax-trick

Jiayi Fu[†], **Xuandong Zhao**[‡], **Ruihan Yang**[♠],
Yuansen Zhang[†], **Jiangjie Chen**[†], **Yanghua Xiao**[†*]
[†]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[♠]School of Data Science, Fudan University
[‡]University of California, Santa Barbara
{fujy22,rhyang21,zhangys22}@m.fudan.edu.cn,
xuandongzhao@ucsb.edu, {jjchen19,shawyh}@fudan.edu.cn

Abstract

Large language models (LLMs) excellently generate human-like text, but also raise concerns about misuse in fake news and academic dishonesty. Decoding-based watermark, particularly the GumbelMax-trick-based watermark (GM watermark), is a standout solution for safeguarding machine-generated texts due to its notable detectability. However, GM watermark encounters a major challenge with generation diversity, always yielding identical outputs for the same prompt, negatively impacting generation diversity and user experience. To overcome this limitation, we propose a new type of GM watermark, the Logits-Addition watermark, and its three variants, specifically designed to enhance diversity. Among these, the GumbelSoft watermark (a softmax variant of the Logits-Addition watermark) demonstrates superior performance in high diversity settings, with its AUROC score outperforming those of the two alternative variants by 0.1 to 0.3 and surpassing other decoding-based watermarking methods by a minimum of 0.1.[1]

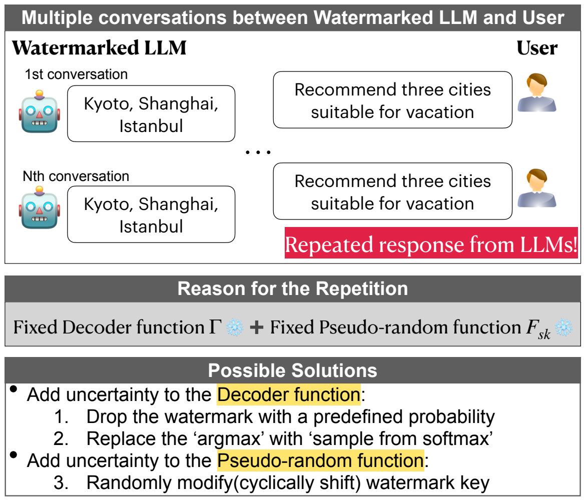

Figure 1: One significant limitation of GM watermark lies in their production of identical responses to the same queries. Such determinism can lead to user dissatisfaction, as individuals may become frustrated with LLM recommending the same outcomes for repeated prompts. This issue primarily stems from the deterministic nature of both the Pseudo-random function and the Decoder function. To address this concern, we propose three solutions: Solutions I and II aim to introduce variability into the Decoder function, whereas Solution III seeks to inject uncertainty into the Pseudo-random function.

1 Introduction

The emergence of large language models (LLMs), exemplified by GPT-4 (OpenAI, 2023a), has enabled the generation of remarkably human-like content, facilitating tasks such as writing (Shanahan and Clarke, 2023), coding (Chen et al., 2021), and fostering creativity. However, this technological advancement brings forth the potential for malicious applications, including social engineering (Mirsky et al., 2023), fake news fabrication (Zellers et al., 2019), and academic dishonesty. Consequently, the need for effective detection of machine-generated texts has become increasingly critical.

Various strategies have been proposed to distinguish machine-generated texts from human-written texts, and decoding-based watermarking has emerged as a highly effective approach. This technique embeds subtle patterns into the text during the decoding stage of LLM, which designated algorithms can identify. The GumbelMax-trick-based watermark (GM watermark), introduced by Aaronson and Kirchner (2023) as their Exponential watermark, is a prominent example within this category, known for its exceptional detectability and low perplexity for generated text. However, a critical limitation of this method is its tendency to produce identical outputs for the same prompt, which could adversely affect both the diversity of the model's outputs and the overall user experience, as illustrated in Figure 1.

*Corresponding author.
[1]Code is available at https://github.com/PorUna-byte/Gumbelsoft

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5791–5808
August 11-16, 2024 ©2024 Association for Computational Linguistics

To address the challenge of generating diverse outputs of the GM watermark, our analysis delves into the core mechanism of decoding-based watermarks. We discover that these watermarks share a cohesive framework, as illustrated in Figure 3. The primary cause of uniform completions for identical prompts is traced back to the deterministic nature of both the *Decoder* and *Pseudo-random* functions in the GM watermark. To mitigate this, we propose two strategies to introduce variability into the Decoder function and one strategy to the Pseudo-random function: 1) Implement a drop mechanism with a predefined probability d_p, enabling direct sampling from the language model without watermark insertion. 2) Replace the "argmax" operation in GumbelMax watermark with "sampling from softmax" with temperature τ. 3) Adjust the watermark key, derived from the Pseudo-random function, by cyclically shifting it r positions—a method to effectively randomize the watermark key.

A critical aspect of this exploration is balancing detectability with diversity. Integrating a dropout probability and shifting the watermark key boosts diversity but also reduces detectability. We propose replacing the argmax operation with "sampling from softmax" to enhance diversity without significantly compromising the watermark's integrity. This approach ensures that even though selections diverge from "argmax", they still achieve high per-token scores, preserving the statistical foundation of the watermark. Further investigation into GM watermark leads us to question the necessity for an exponential transformation in the GumbelMax-trick for embedding watermarks, a technique outlined by Aaronson and Kirchner (2023). Instead, we employ the GumbelMax-trick directly for watermark embedding and propose a distinct type of GM watermark, termed the Logits-Addition watermark.

Our experiments reveal that the GumbelSoft watermark, the softmax variant of the Logits-Addition watermark, consistently outperforms other GM watermark diversified variants in the AUROC metric, achieving a margin of 0.1 to 0.3 in high diversity settings. Additionally, the GumbelSoft watermark surpasses other decoding-based watermarks in AUROC by at least 0.1 on QA tasks, while maintaining low perplexity.

For a clearer understanding of these findings, we have illustrated the relationships among the GumbelMax-trick, the GM watermark (including Exponential and Logits-Addition), and their diversified variants in Figure 2. In conclusion, our contributions are threefold:

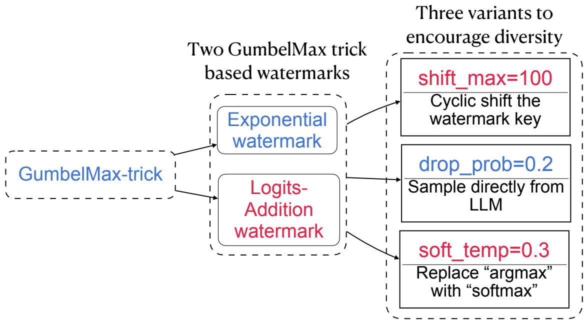

Figure 2: GumbelMax-trick can be used in text watermarking via two different ways: Exponential and Logits-Addition watermark. Each watermark has three variants to enhance generation diversity. The red part denotes our contribution, and the softmax variant of the Logits-Addition watermark is our suggested GumbelSoft watermark.

- We identify the deterministic nature of the Pseudo-random and Decoder functions as the primary cause behind GM watermark producing identical completions for the same prompts and provide a universal framework for all decoding-based watermarking techniques.
- We propose the Logits-Addition watermark as a new type within the GM watermark suite and analyze the expectation and variance for the per-token score. Additionally, we introduce three variants of GM watermark aimed at enhancing the diversity of generated content.
- Our experiments with three varied GM watermark versions reveal that the GumbelSoft watermark surpasses the others in diversity and detectability. Furthermore, our comparative analyses with other decoding-based watermarks show that the GM watermark offers superior detectability and robustness while maintaining quality on par with existing methods.

2 Related Work

Machine-generated text detection can be roughly categorized into three approaches.

Zero-shot Methods. This approach, known as "model self-detection," necessitates full access to the language model and utilizes statistical measures such as perplexity and entropy. Noteworthy contributions include Gehrmann et al. (2019)'s GLTR, Vasilatos et al. (2023)'s perplexity analysis, and Yang et al. (2023a)'s N-gram overlaps.

Mitchell et al. (2023) introduced a perturbation-based method, while Deng et al. (2023) proposed a Bayesian surrogate model. Recent studies encompass Krishna et al. (2023), who advocate for retrieval against paraphrase attacks, and Su et al. (2023), who leverage log-rank ratios. Additionally, Solaiman et al. (2019) employ log probability, Bao et al. (2023) focus on conditional probability curvature, and Venkatraman et al. (2023) utilize uniform information density for improved detection. The primary limitation of zero-shot methods lies in their requirement for complete access to the language model.

Training-Based Methods. These involve classifiers trained to distinguish between machine-generated and human-written texts. Chen et al. (2023); Liu et al. (2023c) use a fine-tuned RoBERTa model (Liu et al., 2019), while Mireshghallah et al. (2023) advocate for partially trained models. Some researchers also use shallow classifiers with extracted text features. OpenAI (2023b); Tian (2023) training classifiers from mixed sources, Yin et al. (2023) using graph structures and contrastive learning, and Tian et al. (2023) applying positive unlabeled training for classifier development. (Li et al., 2023; Tulchinskii et al., 2023). A drawback of training-based methods is their potential over-fitting to specific datasets and models.

Watermarking Techniques. Recent advancements in hidden signal watermarking in texts can be categorized into post-edited and decoding-based watermarking. Post-edited watermarking involves text formatting or lexical changes (Brassil et al., 2002; Sato et al., 2023; He et al., 2022; Yoo et al., 2023a), while decoding-based watermarking in the era of large language models (LLMs) embeds statistical signals during decoding. Notable techniques in this domain include Kirchenbauer et al. (2023)'s red-green list and Zhao et al. (2023)'s robust watermarking. Unbiased watermarks, which preserve the original token distributions, have been explored by Kuditipudi et al. (2023) and Hu et al. (2023). Additionally, multi-bit watermarking, which embeds complex information, is examined by Wang et al. (2023) and Yoo et al. (2023b). Several techniques for embedding watermarks include text formatting, as demonstrated by Por et al. (2012) and Rizzo et al. (2016), and context-aware lexical substitution, as explored by Yang et al. (2021). Syntactic modifications are discussed by Atallah et al. (2001) and Meral et al. (2009). Training data watermarking is addressed by Liu et al. (2023b) and Tang et al. (2023). A publicly detectable watermark is proposed by Fairoze et al. (2023), while leveraging semantic meaning for robustness is examined by Ren et al. (2023). Zhao et al. (2024) propose PF-Watermark to further improve the text perplexity. There are also some surveys on machine-generated content detection (Wu et al., 2023a; Yang et al., 2023b) and text watermarking (Liu et al., 2024).

3 Method

In this section, we will first provide an overview of the decoding-based watermark framework and the GumbelMax-trick. Following this, we'll delve into the application of the GumbelMax-trick in text watermarking and examine their limitations. Concluding the section, we will present our recommended watermark scheme, specifically crafted to overcome these identified limitations.

3.1 Preliminaries

Decoding-Based Watermark Framework. We introduce a concise watermark framework with two main components: the Watermark Generator and Detector, building upon the architecture outlined in Fernandez et al. (2023) and incorporating mathematical concepts from Kuditipudi et al. (2023); Christ et al. (2023). Figure 3 and Table 1 detail the framework's structure and notations.

GumbelMax-trick. The GumbelMax-trick, as proposed by Gumbel (1954), presents an efficient method for sampling from a categorical distribution. Consider a vector of logits $l = (l_1, \ldots, l_K)$ coupled with a sequence of Gumbel-distributed random variables $g_1, \ldots, g_K \sim \text{Gumbel}(0, 1)$. A sample from the categorical distribution $\pi = (\pi_1, \ldots, \pi_K) = \text{softmax}(l_1, \ldots, l_K)$ can be obtained as follows: $w = \arg\max_i (g_i + l_i)$. This sampling approach is referred to as the GumbelMax-trick. It can be demonstrated that this trick is mathematically equivalent to drawing a sample directly from the categorical distribution π, as detailed in the Appendix B.1.

3.2 Watermark Design

Unbiasedness. The GumbelMax-trick enables the creation of an unbiased watermark, which is indistinguishable from unwatermarked text, provided the watermark key's distribution is properly chosen. An unbiased watermark meets the following

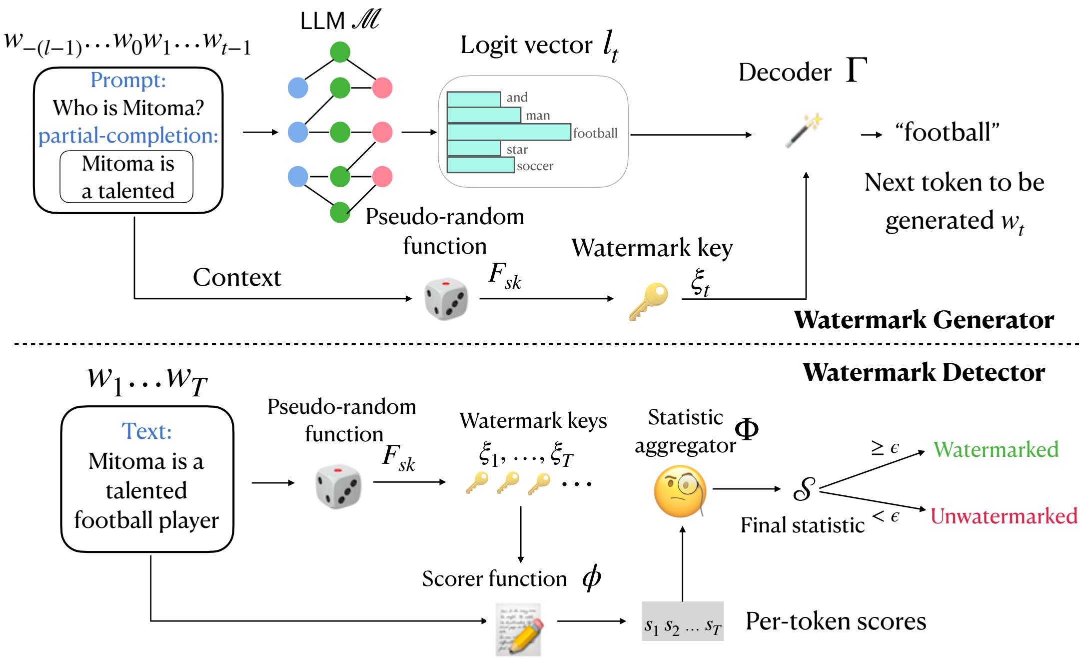

Figure 3: General framework of decoding-based watermark. The Generator uses logits vector l_t and watermark key ξ_t to decode the next token w_t. The Detector, employing scorer ϕ, assesses the correlation between watermark key ξ_t and token w_t, then combines these per-token scores to determine watermark presence. Both Generator and Detector share the same pseudo-random function F_{sk}. The context for watermark key calculation can be the preceding h tokens.

Symbol	**Meaning**
$\mathcal{V}$	Vocabulary, the set of tokens
w_t	Token at position t
$W_g : \mathcal{V}^* \to \mathcal{V}^*$	Watermark Generator, generate a watermarked completion for a given prompt
$\mathcal{D} : \mathcal{V}^* \to \{\text{True, False}\}$	Watermark Detector, detect whether a text is watermarked or not
$l_t \in \mathbb{R}^{\lvert\mathcal{V}\rvert}$	Logits vector for position t, produced by language model $\mathcal{M}$
$\mathcal{M} : \mathcal{V}^* \to \mathbb{R}^{\lvert\mathcal{V}\rvert}$	Language model, give the logits vector l_t for position t based on a proceeding tokens
Ξ	Watermark key space, the set of all possible watermark keys
$\xi_t \in \Xi$	Watermark key at position t
$\mathcal{C}$	Context space, the set of all possible contexts
$F_{sk} : \mathcal{C} \to \Xi$	Pseudo-random function, calculate the watermark key ξ_t
$\Gamma : \mathbb{R}^{\lvert\mathcal{V}\rvert} \times \Xi \to \mathcal{V}$	Decoder function, decode the next token w_t from logits vector and watermark key
$\phi : \mathcal{V} \times \Xi \to \mathbb{R}$	Scorer function, calculate per-token score s_t for each token
$\Phi : \mathbb{R}^* \to \mathbb{R}$	Statistic aggregator, compile all per-token scores into one final statistic

Table 1: Summary of notations.

conditions:

$$\mathbb{P}_{\xi \sim \tau(\cdot)}[\Gamma(\xi, l) = x] = p_x, \forall x \in \mathcal{V}$$

where p is the softmax of l and $\tau(.)$ denotes the watermark key ξ's distribution. Watermark schemes in Aaronson and Kirchner (2023); Wu et al. (2023b); Kuditipudi et al. (2023) are unbiased, unlike the biased method of Kirchenbauer et al. (2023).

Logits-Addition Watermark. The first attempt to use GumbelMax-trick in text watermarking is Aaronson and Kirchner (2023)'s Exponential watermark, which generates subsequent tokens using the formula $w_t = \arg\max_i \frac{\log \xi_t[i]}{p_t[i]}$, where $\xi_t \sim \text{Uniform}(0,1)^{\lvert\mathcal{V}\rvert}$ and $p_t = \text{softmax}(l_t)$. Its detection mechanism computes a per-token score $s_t = -\log(1 - \xi_t[w_t])$.

While the Exponential watermark is linked to empirical entropy, we question the relevance of this connection given that empirical entropy does not accurately reflect the true entropy of the next-token distribution provided by the language model. Consequently, we introduce a new type of GM watermark that directly incorporates Gumbel noise into the logits vector for next-token sampling: $w_t = \arg\max_i (l_t[i] + \xi_t[i])$, where $\xi_t \sim \text{Gumbel}(0,1)^{|\mathcal{V}|}$ and l_t represents the logit vector. This method's detection algorithm calculates a per-token score $s_t = \xi_t[w_t]$, a technique we designate as the Logits-Addition Watermark.

We assert that despite the token generation processes of these two methods being equivalent (see Appendix B.2), their detection mechanisms differ. Furthermore, the softmax variant of our Logits-Addition watermark demonstrates superior diversity and detectability compared to the Exponential watermark's softmax variant (refer to Figure 4). This supports our rationale for applying Gumbel noise directly and adopting an alternative detection method. Moreover, we present a theorem detailing the expectation and variance of the per-token score within the Logits-Addition watermark.

Theorem 1. *Consider a text $w_1, \ldots, w_T$ embedded with a watermark using the Logits-Addition technique. When evaluated by the Logits-Addition watermark detector, the expected value and variance of the score for each token are given by*

$$\begin{aligned}\mathbb{E}[s_t] =& \mathbb{E}[\xi_t[w_t]] = -\log(p_t[w_t]) + \gamma, \\ Var\,[s_t] \leq& \frac{2p_t\,[w_t]^2}{(1-p_t\,[w_t])^3} + \frac{2}{p_t\,[w_t]} \\ &- (-\log p_t\,[w_t] + \gamma)^2\,.\end{aligned}$$

For a non-watermarked text $w_1, \ldots, w_T$, applying the Logits-Addition watermark detector, the expected value and variance for each per-token score are

$$\begin{aligned}\mathbb{E}[s_t] &= \mathbb{E}[\xi_t[w_t]] = \gamma, \\ Var[s_t] &= Var[\xi_t[w_t]] = \frac{\pi^2}{6}.\end{aligned}$$

Here, γ denotes the Euler-Mascheroni constant, and $p_t = softmax(l_t)$, is derived from the language model.

The proof for this theorem can be found in Appendix B.3. According to this theorem, if certain watermarked tokens are assigned a low probability by the language model, the expectation of their per-token scores, given by $-\log(p_t[w_t]) + \gamma$, significantly increases. This makes these tokens notably easier to detect.

Limitations of the GM Watermark. Despite its effectiveness in watermarking texts, the GumbelMax-trick has limitations. One major limitation is that it generates deterministic outputs, resulting in identical completions for the same prompts (as shown in Figure 1). Such determinism can lead to user dissatisfaction, as individuals may become frustrated with LLM consistently recommending the same outcomes for the same queries. To address this issue and improve output diversity, we propose three diversified GM watermark variants. These variants are thoroughly outlined in the Introduction section (see Section 1) and are aimed at enhancing the diversity of the generation process.

3.3 GumbelSoft Watermark

Algorithm 1 GumbelSoft Generator

Input: prompt x, LLM $\mathcal{M}$, temperature τ.
Output: Watermarked completion $w_1, \ldots, w_T$

for $t = 1, \ldots, T$ **do**
    Logits $l_t \leftarrow \mathcal{M}(x, w_{1,\ldots,t-1})$
    Watermark key ξ_t ←hash context to a Gumbel-distributed vector
    w_t ←sample from softmax$((\xi_t + l_t)/\tau)$
end for
return $[w_1, \ldots, w_T]$

Algorithm 2 GumbelSoft Detector

Input: Text input $w_{1,\ldots,T}$; a predefined threshold ϵ
Output: Boolean indicator: True if watermark detected, False otherwise

for $t = 1, \ldots, T$ **do**
    Watermark key ξ_t ←hash context to a Gumbel-distributed vector
    Per-token score $s_t \leftarrow \xi_t[w_t]$
end for
Calculate Final statistic $\mathcal{S}$:

$$\mathcal{S} = \Phi(s_1, s_2, \ldots, s_T) = \frac{\sqrt{6T}}{\pi}\left(\frac{\sum_{i=1}^{T} s_i}{T} - \gamma\right)$$

with $\gamma \approx 0.5772$ denoting the Euler-Mascheroni constant.
return True if $\mathcal{S} \geq \epsilon$ else False.

After conducting a comprehensive series of experiments with three diversified variants of both the Exponential and Logits-Addition watermarks, we identified the GumbelSoft watermark as the most effective, achieving Pareto optimality. The methodologies for both the Generator and Detector of the GumbelSoft watermark are elaborated in Algorithms 1 and 2, respectively.

We now explain the key insight behind the GumbelSoft watermark. The Logits-Addition watermark is primarily characterized by differing expected per-token scores for watermarked and unwatermarked texts. Leveraging this difference allows for the construction of a detection mechanism based on the null hypothesis $\mathcal{H}_0$: *The text is unwatermarked*. Following the z-test by Kirchenbauer et al. (2023), we devise the final statistic $\mathcal{S}$ of Logits-Addition watermark to be:

$$\mathcal{S} = \Phi(s_1, s_2, \ldots, s_T) = \frac{\sqrt{6T}}{\pi}\left(\frac{\sum_{i=1}^{T} s_i}{T} - \gamma\right)$$

According to expectation Theorem 1 and the central limit Theorem (Fischer, 2011), we notice that for unwatermarked texts, $\mathcal{S}$ aligns with a standard Gaussian distribution. In contrast, for watermarked texts, $\mathcal{S}$ deviates, typically presenting significantly higher values. Given that GumbelSoft is a variant of Logits-Addition, it naturally inherits its characteristics. Consequently, the majority of tokens sampled by the GumbelSoft watermark are likely identical to those selected by the Logits-Addition watermark. Moreover, tokens not usually favored by Logits-Addition are observed to have comparatively higher per-token scores.

4 Experiment

This section presents a comparative study of three diversified variants (refer to Figure 1) of both Exponential and Logits-Addition watermarks, with an emphasis on aspects such as detectability, diversity, and quality. Following this, the optimal diversified variant of the GM watermark, the GumbelSoft watermark, is identified and compared against several existing decoding-based watermark schemes (see Appendix A for details).

4.1 Experimental Setting

We briefly outline our experimental setup, including the datasets, models, metrics, and baselines used, specifically for the Completion and QA tasks.

Dataset and Models. In our experimental setup, each task employs unique language models and datasets. For the Completion task, the Llama2-7b model (Touvron et al., 2023) and C4 dataset (Raffel et al., 2019) are used to assess detectability, while diversity is evaluated through 20 high-entropy prompts repeated 50 times each on Llama2-7b. Perplexity is calculated using Llama2-13b with the C4 dataset. For the QA task, we utilize the Llama2-7b-chat model and Alpaca dataset (Taori et al., 2023) for detectability, and assess diversity with 20 chat-like prompts on Llama2-7b-chat, also repeated 50 times. Perplexity here is measured using Llama2-13b-chat on the Alpaca dataset.

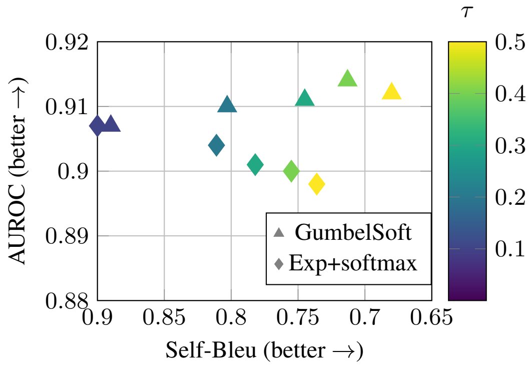

Figure 4: The figure shows how AUROC changes with Self-Bleu on the QA task. we use different colors to represent temperature and different marks to represent GumbelSoft and the softmax variant of Exponential watermarks. The AUROC is calculated for 100 detection tokens. Since the top-right outshines the bottom-left in performance, GumbelSoft is more effective than the softmax variant of Exponential.

Metrics. Our detectability evaluation relies on AUROC, FPR at a fixed FNR of 0.01, and FNR at a fixed FPR of 0.01. We assess generation quality using perplexity, derived from a larger model. To measure generation diversity, our approach includes Self-BLEU and Distinct 1-gram and 2-gram.

Baselines. The universal decoding-based watermark framework, as presented in Figure 3, serves to categorize all decoding-based watermark schemes, including those proposed by Kirchenbauer et al. (2023); Aaronson and Kirchner (2023); Wu et al. (2023b); Kuditipudi et al. (2023). These schemes are the baselines in our study. Their mathematical representations, provided in Appendix A, illustrate their integration into our unified taxonomy.

4.2 Diversity

This subsection aims to identify which variant of the two GM watermarks is best in terms of diversity and detectability. A detailed comparison of our GumbelSoft watermark with other GM watermark variants in the QA task is presented in Table 2, with results for the Completion task detailed in Appendix C.1. These results indicate that our GumbelSoft method achieves superior content di-

		Diversity			Detectability			Quality
		Self-Bleu ↓	Dist-1 ↑	Dist-2 ↑	AUROC ↑	FPR ↓	FNR ↓	PPL ↓
Exponential	vanilla	1.000	0.011	0.017	0.905	0.749	0.569	1.985
	drop_prob=0.10	0.852	0.070	0.196	0.891	0.790	0.623	2.020
	drop_prob=0.20	0.767	0.087	0.261	0.871	0.835	0.691	2.015
	drop_prob=0.30	0.715	0.097	0.298	0.845	0.870	0.752	2.077
	drop_prob=0.40	**0.676**	0.103	0.325	0.816	0.896	0.808	2.000
	shift_max=30	0.902	0.080	0.227	0.742	0.946	0.825	1.996
	shift_max=50	0.839	0.090	0.266	0.700	0.963	0.882	1.985
	shift_max=100	0.741	0.101	0.311	0.672	0.963	0.900	1.982
	shift_max=200	0.689	**0.106**	**0.331**	0.644	0.970	0.901	1.983
	soft_temp=0.2	0.811	0.084	0.233	0.904	0.748	0.586	2.372
	soft_temp=0.3	0.782	0.087	0.254	0.901	0.756	0.597	2.096
	soft_temp=0.4	0.755	0.094	0.276	0.900	0.794	0.598	2.239
	soft_temp=0.5	0.736	0.096	0.288	0.898	0.798	0.602	2.127
Logits-Addition	vanilla	1.000	0.011	0.017	0.908	0.743	0.579	1.985
	drop_prob=0.10	0.823	0.074	0.212	0.887	0.769	0.634	1.998
	drop_prob=0.20	0.762	0.089	0.263	0.867	0.830	0.701	1.994
	drop_prob=0.30	0.713	0.097	0.300	0.846	0.833	0.748	2.088
	drop_prob=0.40	0.691	0.102	0.316	0.810	0.888	0.808	1.988
	shift_max=30	0.906	0.080	0.224	0.730	0.961	0.838	1.986
	shift_max=50	0.824	0.092	0.272	0.694	0.965	0.886	1.986
	shift_max=100	0.751	0.101	0.309	0.670	0.971	0.903	1.981
	shift_max=200	0.694	**0.106**	**0.331**	0.642	0.981	0.917	1.981
	soft_temp=0.2	0.803	0.083	0.235	0.910	0.726	**0.568**	2.338
	soft_temp=0.3	0.745	0.095	0.281	0.911	**0.704**	0.572	2.027
	soft_temp=0.4	0.713	0.098	0.300	**0.914**	0.713	0.570	2.169
	soft_temp=0.5	0.680	0.105	0.326	0.912	0.742	0.571	2.221

Table 2: Comparison of three diversified variants of both Exponential and Logits-Addition watermarks in the QA task. These variants include **drop_prob=0.2**, sampling from the language model directly at a 0.2 probability; **shift_max=100**, where the watermark key is cyclically shifted within a 0-100 range; and **soft_temp=0.3**, which uses a softmax sampling with a temperature of 0.3 to balance randomness. **Vanilla** is the original GM watermark(Exponential and Logits-Addition) without any technique to enhance diversity. The detectability is measured by 100 detection tokens. Note that Logits-Addition+soft_temp is the GumbelSoft watermark. GumbelSoft is the best of three diversified variants of GM watermark in terms of both detectability and diversity.

versity and detectability compared to other variants, though it incurs a slight increase in perplexity. We also notice that the GumbelSoft watermark is better than the softmax variant of the Exponential watermark under the same temperature setting, which is clearly shown in Figure 4.

While methods like drop probability and watermark key shift can enhance diversity, they tend to negatively impact detectability. The decrease in detectability due to drop probability may be attributed to a fraction of tokens not being sampled using the watermark key, thereby diluting the overall statistical strength. In the case of shifted watermark keys, the detection phase becomes more complex as every possible shift must be tested to identify the watermark, potentially leading to inflated statistics for unwatermarked texts and thus reducing detectability. In contrast, our GumbelSoft watermark does not encounter these issues, maintaining high detectability while also enhancing generation diversity.

4.3 Detectability and Quality

This subsection aims to show that the GumbelSoft watermark is better than other decoding-based watermarks in terms of detectability, the results are shown in Table 3 and the hyperparameter is detailed in Appendix C.2.

GumbelSoft watermark exhibits the highest detectability, likely explained by the expectation and variation theory in Theorem 1. Increased detection token amounts also improve detectability, aligning with findings from Chakraborty et al. (2023). The high-entropy Llama2-7b model in Completion tasks shows greater detectability than the lower entropy Llama2-7b-chat in QA tasks, as high entropy facilitates easier watermark embedding. Regarding generation quality (perplexity), GumbelSoft shows

		# tokens=40			# tokens=60			# tokens=100			
		AUROC ↑	**FPR ↓**	**FNR ↓**	**AUROC ↑**	**FPR ↓**	**FNR ↓**	**AUROC ↑**	**FPR ↓**	**FNR ↓**	**PPL ↓**
Completion	Unwatermarked	-	-	-	-	-	-	-	-	-	11.576
	KGW	0.970	0.616	0.361	0.988	0.329	0.164	0.997	0.078	0.041	14.217
	Exponential	0.997	0.012	0.012	0.999	**0.000**	0.006	**1.000**	**0.000**	**0.000**	10.953
	Dipmark	0.935	0.693	0.565	0.968	0.483	0.362	0.988	0.274	0.153	**8.664**
	ITS	0.961	0.073	1.000	0.978	0.040	1.000	0.994	0.010	0.402	11.843
	GumbelSoft	**0.998**	**0.011**	**0.010**	**1.000**	**0.000**	**0.005**	**1.000**	**0.000**	0.001	11.820
QA	Unwatermarked	-	-	-	-	-	-	-	-	-	1.980
	KGW	0.657	0.985	0.969	0.701	0.978	0.945	0.754	0.949	0.901	2.081
	Exponential	0.780	0.892	0.813	0.840	0.852	0.738	0.905	0.749	**0.569**	1.985
	Dipmark	0.588	0.988	0.982	0.615	0.981	0.984	0.646	0.979	0.970	**1.792**
	ITS	0.583	1.000	1.000	0.618	0.963	1.000	0.665	0.954	1.000	2.011
	GumbelSoft	**0.788**	**0.866**	**0.812**	**0.848**	**0.837**	**0.722**	**0.911**	**0.704**	0.572	2.027

Table 3: A comparative analysis of the detectability across various decoding-based watermarking schemes. Detectability is assessed for varying token counts: 40, 60, and 100. The temperature for GumbelSoft is set to 0.3. GumbelSoft shows high detectability and low perplexity compared with other decoding-based watermarks.

Figure 5: Comparison of the robustness of decoding-based watermark on Completion task. Blue histograms indicate unattacked conditions and red histograms show attacked scenarios. The AUROC is calculated for 40 detection tokens, with GumbelSoft set at a 0.3 temperature. **Exp**, **Dip**, and **GS** refer to Exponential, Dipmark, and GumbelSoft, respectively. GumbelSoft and Exponential show higher robustness when facing the T5-span attack.

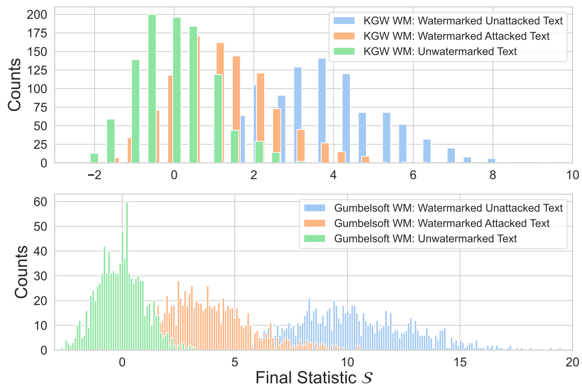

Figure 6: Comparison of final statistic for KGW and GumbelSoft watermark on Completion task. The final statistic is calculated for 40 detection tokens, with GumbelSoft set at the temperature of 0.3. The robustness of GumbelSoft stems from the strong pattern of the GM watermark, ensuring a large gap in the Final statistic between watermarked text and natural text.

relatively low perplexity. In contrast, the KGW watermark's biased logits modification leads to high perplexity, while Dipmark's strategy of amplifying high-probability tokens results in the lowest perplexity in the Completion task. For the QA task, the low perplexity across all methods, attributed to the low entropy of Llama2-7b-chat, diminishes the value of comparative perplexity analysis.

4.4 Robustness

In this section, we assess the robustness of various decoding-based watermarking schemes, with results for the Completion task in Figures 5 and for the QA task in Appendix C.3. All texts, both watermarked and unwatermarked, were tested under the T5-span attack (explained in Appendix C.3).

Our key finding reveals that the Exponential and GumbelSoft watermarks are particularly robust against the T5-span attack, in contrast to other watermarks. Their AUROC values, as well as FPR and FNR metrics, remained stable post-attack, while other schemes experienced significant declines. This robustness can be attributed to the effective embedding of watermarks by the GumbelMax-trick, ensuring significant final statistics despite per-token score alterations. Further analysis, involving a comparative study of final statistic distributions between KGW and GumbelSoft watermarks, is shown in Figure 6. The results demonstrate that while attacked watermarked texts under KGW show considerable overlap with unwatermarked texts, our GumbelSoft watermark displays less overlap, indicating its greater robustness.

5 Conclusion

We observed that the GumbelMax-trick-based watermark(GM watermark) produces identical responses to identical queries due to the deterministic nature of both the Decoder and Pseudo-random functions. To address this, we introduce three diversified variants aimed at enhancing GM watermark diversity. Furthermore, we question the need for an Exponential transformation (Aaronson and Kirchner, 2023) in watermark embedding and propose a new approach named Logits-Addition watermark. Our experiments across these variants for both Exponential and Logits-Addition watermarks identified GumbelSoft, a softmax-based Logits-Addition variant, as the optimal choice. Comparative analysis with other decoding-based watermarks demonstrated that GumbelSoft surpasses in detectability, maintains lower perplexity, and ensures higher robustness.

Limitations

GumbelSoft watermark's Ngram pseudo-random function is susceptible to paraphrase attacks due to its dependence on the previous h tokens for key determination. In terms of quality assessment, we solely rely on perplexity, whereas some studies utilize downstream tasks for evaluation. Our mathematical analysis is focused solely on the Logits-Addition watermarking technique, we do not provide a comprehensive mathematical analysis of the GumbelSoft watermark.

Ethical Considerations

As advanced language models increasingly demonstrate remarkable capabilities, concerns regarding their misuse have escalated. Consequently, the development of effective methods for detecting machine-generated text has become crucial. The GM watermark has emerged as a highly effective technique for differentiating between machine-generated and natural text. Nevertheless, the GM watermark is limited by issues of diversity, which may hinder its practical application. The GumbelSoft watermark represents a straightforward yet effective strategy to address this limitation. This approach maintains the watermark's detectability while significantly enhancing its generative diversity. We believe that our method will facilitate the broader implementation of the GM watermark in various applications.

Acknowledgement

We would like to express our gratitude to our colleagues at Fudan University, especially to Yikai Zhang, Caiyu Hu, and Wei Shi for their valuable comments and spirited discussions on this project. This research is funded by the Science and Technology Commission of Shanghai Municipality Grant (No. 22511105902)

References

S. Aaronson and H. Kirchner. 2023. Watermarking gpt outputs. Technical report, openai.

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 *Proceedings of the 4th International Workshop on Information Hiding*, IHW '01, page 185–199, Berlin, Heidelberg. Springer-Verlag.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature.

J. Brassil, S. Low, N. Maxemchuk, and L. O'Gorman. 2002. Electronic marking and identification techniques to discourage document copying. In *Proceedings of INFOCOM '94 Conference on Computer Communications*.

Souradip Chakraborty, AmritSingh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023. On the possibilities of ai-generated text detection. *arXiv: Learning*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Gpt-sentinel: Distinguishing human and chatgpt generated content.

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable watermarks for language models.

Zhijie Deng, Hongcheng Gao, Yibo Miao, and Hao Zhang. 2023. Efficient detection of llm-generated texts with a bayesian surrogate model.

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. 2023. Publicly detectable watermarking for language models. Cryptology ePrint Archive, Paper 2023/1661. https://eprint.iacr.org/2023/1661.

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

Hans Fischer. 2011. *A History of the Central Limit Theorem*. New York: Springer.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. Gltr: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

E.J. Gumbel. 1954. Statistical theory of extreme values and some practical applications: A series of lectures. *U.S. Government Printing Office eBooks,U.S. Government Printing Office eBooks*.

Xuanli He, Qiongkai Xu, Yi Zeng, Lingjuan Lyu, Fangzhao Wu, Jiwei Li, and Ruoxi Jia. 2022. Cater: Intellectual property protection on text generation apis via conditional watermarks.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2023. Unbiased watermark for large language models.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense.

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

Linyang Li, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. 2023. Origin tracing and detecting of llms.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023a. A semantic invariant robust watermark for large language models.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip S. Yu. 2024. A survey of text watermarking in the era of large language models.

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. *Cornell University - arXiv,Cornell University - arXiv*.

Yixin Liu, Hongsheng Hu, Xun Chen, Xuyun Zhang, and Lichao Sun. 2023b. Watermarking classification dataset for copyright protection.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023c. Check me if you can: Detecting chatgpt-generated academic writing using checkgpt.

Hasan Mesut Meral, Bülent Sankur, A. Sumru Özsoy, Tunga Güngör, and Emre Sevinç. 2009. Natural language watermarking via morphosyntactic alterations. *Computer Speech & Language*, page 107–125.

Fatemehsadat Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. 2023. Smaller language models are better black-box machine-generated text detectors.

Yisroel Mirsky, Ambra Demontis, Jaidip Kotak, Ram Shankar, Deng Gelei, Liu Yang, Xiangyu Zhang, Maura Pintor, Wenke Lee, Yuval Elovici, and Battista Biggio. 2023. The threat of offensive ai to organizations. *Computers & Security*, page 103006.

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

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

OpenAI. 2023b. New ai classifier for indicating ai-written text. Technical report, openai.

Lip Yee Por, KokSheik Wong, and Kok Onn Chee. 2012. Unispach: A text-based data hiding method using unicode space characters. *Journal of Systems and Software*, 85(5):1075–1082.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and PeterJ. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv: Learning*.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2023. A robust semantics-based watermark for large language model against paraphrasing.

Stefano Giovanni Rizzo, Flavio Bertini, and Danilo Montesi. 2016. Content-preserving text watermarking through unicode homoglyph substitution. In *Proceedings of the 20th International Database Engineering & Applications Symposium on - IDEAS '16*.

Ryoma Sato, Yuki Takezawa, Han Bao, Kenta Niwa, and Makoto Yamada. 2023. Embarrassingly simple text watermarks.

Murray Shanahan and Catherine Clarke. 2023. Evaluating large language model creativity from a literary perspective.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text.

Ruixiang Tang, Qizhang Feng, Ninghao Liu, Fan Yang, and Xia Hu. 2023. Did you train on my dataset? towards public dataset protection with clean-label backdoor watermarking.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Edward Tian. 2023. More than an ai detector preserve what's human. Technical report, Princeton University.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, Qinghua Zhang, Ruifeng Li, Chao Xu, and Yunhe Wang. 2023. Multiscale positive-unlabeled detection of ai-generated texts.

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.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Serguei Barannikov, Irina Piontkovskaya, Sergey Nikolenko, and Evgeny Burnaev. 2023. Intrinsic dimension estimation for robust detection of ai-generated texts.

Christoforos Vasilatos, Manaar Alam, Talal Rahwan, Yasir Zaki, and Michail Maniatakos. 2023. Howkgpt: Investigating the detection of chatgpt-generated university student homework through context-aware perplexity analysis.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2023. Gpt-who: An information density-based machine-generated text detector.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Towards codable watermarking for injecting multi-bit information to llm.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F. Wong, and Lidia S. Chao. 2023a. A survey on llm-generated text detection: Necessity, methods, and future directions.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. 2023b. Dipmark: A stealthy, efficient and resilient watermark for large language models.

Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. 2021. Tracing text provenance via context-aware lexical substitution.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023a. Dnagpt: Divergent n-gram analysis for training-free detection of gpt-generated text.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023b. A survey on detection of llms-generated content.

Nan Yin, Li Shen, Mengzhu Wang, Long Lan, Zeyu Ma, Chong Chen, Xian-Sheng Hua, and Xiao Luo. 2023. Coco: A coupled contrastive framework for unsupervised domain adaptive graph classification.

KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. 2023a. Robust multi-bit natural language watermarking through invariant features.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. 2023b. Advancing beyond identification: Multi-bit watermark for large language models.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Cornell University - arXiv,Cornell University - arXiv*.

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

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. 2024. Permute-and-flip: An optimally robust and watermarkable decoder for llms.

A Baselines

Here, we present a consolidated mathematical representation within a unified taxonomy for the baseline watermark schemes.

A.1 KGW Scheme

For the KGW scheme, as proposed by Kirchenbauer et al. (2023):

- **Context**: The previous h tokens.
- **Pseudo-random Function**: $F_{sk}(\text{context})$ hashes the context to seed, then uses this seed to generate a random vector in $\{0,1\}^{|\mathcal{V}|}$, the vector has $\gamma|\mathcal{V}|$ 1's and $(1-\gamma)|\mathcal{V}|$ 0's.
- **Decoder**: $\Gamma(\xi_t, l_t)$ samples a token from $\text{softmax}(\delta * \xi_t + l_t)$.
- **Scorer**: $\phi(\xi_t, w_t) = \xi_t[w_t]$.
- **Statistic Aggregator**:

$$\Phi(s_1, \ldots, s_T) = \frac{\sum_{t=1}^{T} s_t - \gamma T}{\sqrt{T\gamma(1-\gamma)}}$$

A.2 Exponential Scheme

For the Exponential scheme, as proposed by Aaronson and Kirchner (2023):

- **Context**: The previous h tokens.
- **Pseudo-random Function**: $F_{sk}(\text{context})$ hashes the context to a seed, then uses this seed to generate a random vector in $(0,1)^{|\mathcal{V}|}$, each element is uniformly sample from (0,1).
- **Decoder**: $\Gamma(\xi_t, l_t) = \arg\max_{1 \le i \le |\mathcal{V}|} \frac{\log \xi_t[i]}{p_t[i]}$, where $p_t = \text{softmax}(l_t)$.
- **Scorer**: $\phi(\xi_t, w_t) = -\log(1 - \xi_t[w_t])$.
- **Statistic Aggregator**:

$$\Phi(s_1, \ldots, s_T) = \frac{1}{\sqrt{T}} \sum_{t=1}^{T} s_t - \sqrt{T}$$

A.3 Dipmark Scheme

For the Dipmark scheme, as proposed by Wu et al. (2023b):

- **Context**: The previous h tokens.
- **Pseudo-random Function**: $F_{sk}(\text{context})$ hashes the context to a seed, then uses this seed to generate a random permutation on the vocabulary $\mathcal{V}$.
- **Decoder**: $\Gamma(\xi_t, l_t)$ samples token $\xi_t[i]$ with probability $\lambda(i) - \lambda(i-1)$, where $\lambda(i) = \max\{\sum_{j=1}^{i} p_t(\xi_t[j]) - \alpha, 0\} + \max\{\sum_{j=1}^{i} p_t(\xi_t[j]) - (1-\alpha), 0\}$, where p_t=softmax(l_t).
- **Scorer**: $\phi(\xi_t, w_t) = \mathbf{1}_{\{w_t \in \xi_t[\gamma|\mathcal{V}|:|\mathcal{V}|]\}}$.
- **Statistic Aggregator**:

$$\Phi(s_1, s_2, \ldots, s_T) = \frac{\sum_{t=1}^{T} s_t - (1-\gamma)T}{\sqrt{T}}$$

A.4 ITS Scheme

For the ITS scheme, as proposed by Kuditipudi et al. (2023):

- **Context**: A global watermark key sequence ξ-list and the position index t. Each watermark key ξ-list[i] consists of a permutation π on the vocabulary and a random number μ in $(0,1)$.
- **Pseudo-random Function**: $F_{sk}(\text{context}) = \xi$-list[t].
- **Decoder**: $\Gamma(\xi_t, l_t) = \pi^{-1}(\min\{\pi(i) : p_t(\{j : \pi(j) \le \pi(i)\}) \ge \mu\})$, where $\xi_t = (\mu, \pi)$ and $p_t = \text{softmax}(l_t)$.
- **Scorer**: $\phi(\xi_t, w_t) = |\mu - \eta(\pi(w_t))|$, where $\eta(k) = \frac{k-1}{|\mathcal{V}|-1}$.
- **Statistic Aggregator**:

$$\Phi(s_1, s_2, \ldots, s_T) = -\frac{1}{T} \sum_{t=1}^{T} s_t$$

A.5 Design Principles

We now explore the design principles underlying these fundamental components.

Context For Watermark generators and detectors, it is essential to recognize that they share the same context, which is constrained to the previous tokens of w_t. This limitation arises from the auto-regressive nature of the Watermark generator, which sequentially generates tokens from left to right. A conventional approach for context selection is to use the previous h tokens:

$w_{t-h}, \ldots, w_{t-1}$. However, this design is vulnerable to paraphrase attacks, as such attacks can alter these preceding tokens, subsequently modifying the watermark key ξ_t, and ultimately affecting the per-token score s_t. A more robust approach involves considering the semantic meaning of previous tokens, based on the rationale that paraphrasing maintains the semantics despite changing the tokens (Liu et al., 2023a). Kuditipudi et al. (2023) suggest utilizing a global list for storing all watermark keys and retrieving a specific watermark key using the position index t

Pseudo-random Function. The pseudo-random function's role is to determine the watermark key ξ_t based on the given context. This function could be as basic as a hash function of the context or might involve leveraging an embedding model to extract the context's semantic content. An alternative method is to use the position index t to retrieve a watermark key from a global list. It is crucial to note that both the Watermark generator and detector share the same pseudo-random function.

Decoder. The decoder is integral to the Watermark generator, utilizing the watermark key ξ_t and the logits vector l_t to determine the subsequent token w_t. Implementation methods for this component vary among different watermarks.

Scorer. The scorer is to establish a correlation between the watermark key ξ_t and the token w_t. Nevertheless, using a global watermark key list and the position index t for key retrieval can result in a significant alignment shift issue. This problem manifests as a misalignment between the watermark key ξ_t and the token w_t in texts subjected to insertion or deletion attacks. To address this, Kuditipudi et al. (2023) recommend using alignment cost or edit distance for computing the sequence score, as opposed to the per-token score.

Statistic Aggregator. Finally, the statistic aggregator compiles all per-token scores or employs a single sequence score to ascertain the presence of a watermark. A typical method involves calculating the z-score and p-value of collected scores. Alternatively, one could use the empirical cumulative distribution function (Kuditipudi et al., 2023) for final statistical analysis.

B Mathematical Proofs

B.1 Unbiasedness for GumbelMax

We demonstrate that the GumbelMax-trick is mathematically equivalent to directly sampling from the categorical distribution π, thereby establishing its unbiased nature when utilized in text watermarking applications.

Denote the vocabulary as $\mathcal{V}$, the vector of logits as $l = (l_1, \ldots, l_{|\mathcal{V}|})$, and a sequence of independent Gumbel-distributed random variables as $\xi_1, \ldots, \xi_{|\mathcal{V}|} \sim \mathrm{Gumbel}(0, 1)$.

$$\mathbb{P}_{\xi \sim \mathrm{Gumbel}(0,1)^{|\mathcal{V}|}} \left[\arg\max_{1 \leq i \leq |\mathcal{V}|} \{\xi_i + l_i\} = x \right] \tag{1}$$

$$= \mathbb{P}_{\eta \sim Q(\cdot)} \left[\arg\max_{1 \leq i \leq |\mathcal{V}|} \eta_i = x \right] \tag{2}$$

$$= \mathbb{P}_{\eta \sim Q(\cdot)} \left[\eta_x \geq \eta_i, \forall i \neq x \right] \tag{3}$$

$$= \mathop{\mathbb{E}}_{Y \sim q(\cdot)} \left[\prod_{i \neq x} \mathbb{P}\left[Y \geq \eta_i \right] \right] \tag{4}$$

$$= \int_{-\infty}^{+\infty} f(y - l_x) \prod_{i \neq x} F(y - l_i) dy \tag{5}$$

$$= \int_{-\infty}^{+\infty} e^{-((y-l_x)+e^{-(y-l_x)})} \prod_{i \neq x} e^{-e^{-(y-l_i)}} dy \tag{6}$$

$$= \int_{-\infty}^{+\infty} e^{\sum_{i=1}^{|\mathcal{V}|} -e^{l_i - y}} e^{l_x - y} dy \tag{7}$$

$$= \int_{-\infty}^{+\infty} e^{-e^{-y} \sum_i e^{l_i}} e^{-y} e^{l_x} dy \tag{8}$$

$$= Z p_x \int_{-\infty}^{+\infty} e^{-Ze^{-y}} e^{-y} dy \tag{9}$$

$$= Z p_x \frac{1}{Z} \tag{10}$$

$$= p_x \tag{11}$$

In equation (2), we introduce a variable substitution $\eta_i = \xi_i + l_i$ for simplification. Moving to equation (4), we define the random variable Y to represent η_x for enhanced clarity, and we assume that Y follows the distribution $q(.)$. Furthermore, we utilize the independence of $\eta_i, i = 1, \ldots, |\mathcal{V}|$. In equation (5), $f(.)$ denotes the probability density function of the Gumbel(0,1) distribution, while $F(.)$ represents its cumulative distribution function. Finally, in equation (9), we introduce $Z = \sum_i e^{l_i}$ as a notation simplification.

B.2 Equivalence of Two Representations

We contend that the token generation processes for the Logits-Addition watermark and the Exponential watermark are mathematically equivalent, though their respective per-token scoring mechanisms differ. To illustrate this equivalence, we present the following equations, equation (12) defines the Logits-Addition watermark, while equation (15) corresponds to the definition of the Exponential watermark.

$$
\begin{aligned}
w &= \arg\max_{1\le i\le |\mathcal{V}|} \{\eta_i + l_i\} && (12)\\
&= \arg\max_{1\le i\le |\mathcal{V}|} e^{\eta_i + l_i} && (13)\\
&= \arg\max_{1\le i\le |\mathcal{V}|} \frac{-p_i}{\log \xi_i} && (14)\\
&= \arg\max_{1\le i\le |\mathcal{V}|} \frac{\log \xi_i}{p_i} && (15)
\end{aligned}
$$

Here, we utilize the relationship $p_i = \text{softmax}(l_i) \propto e^{l_i}$ and $\eta_i = -\log(-\log \xi_i)$. In these notations, we omit the position index t for simplicity, and we assume $\eta_i \sim \text{Gumbel}(0,1)$ and $\xi_i \sim \text{Uniform}(0,1)$.

While the token generation processes for the Logits-Addition watermark and the Exponential watermark are equivalent, their scoring methods are distinct:

$$
\begin{aligned}
w &= \eta[w] && (16)\\
&= -\log(-\log \xi[w]) && (17)\\
&\ne -\log(1-\xi[w]) && (18)
\end{aligned}
$$

Equation (16) specifies the per-token scoring for the Logits-Addition watermark while equation (18) is the scoring method for the Exponential watermark.

B.3 Expectation and Variance for Per-token Score

We now establish the expected per-token score for texts, distinguishing between those with and without the Logits-Addition watermark. In the case of unwatermarked texts, the watermark token, w_t, exhibits no correlation with ξ_t. Consequently, $\xi_t[w_t]$ adheres to a Gumbel(0,1) distribution. This leads to

$$
\mathbb{E}[s_t] = \mathbb{E}[\xi_t[w_t]] = \gamma,
$$

$$
\text{Var}[s_t] = \text{Var}[\xi_t[w_t]] = \frac{\pi^2}{6}.
$$

Conversely, for watermarked texts, a correlation exists between w_t and ξ_t. This correlation alters the distribution of $\xi_t[w_t]$, diverging it from the standard Gumbel(0,1) form. To compute its expected value, we define $\xi_t[w_t]$ as a random variable X. We then deduce its cumulative distribution function (CDF), $F(x)$, and probability density function (PDF), $f(x)$. Utilizing this PDF, we calculate the expectation of X. For simplification, we exclude the position index 't' in the subsequent equations. Here, ξ_i represents $\xi[i]$, implying ξ_w is equivalent to $\xi_t[w_t]$, and similar conventions apply to other notations.

$$
\begin{aligned}
F(x) &= \mathbb{P}\left[X \le x\right] && (19)\\
&= \mathbb{P}\left[\xi_w \le x\right] && (20)\\
&= \mathbb{P}\left[\xi_i + l_i - l_w \le x, \forall i\right] && (21)\\
&= \mathbb{P}\left[e^{\xi_i + l_i - l_w} \le e^x, \forall i\right] && (22)\\
&= \mathbb{P}\left[\frac{-1}{\log h_i}\frac{p_i}{p_w} \le e^x, \forall i\right] && (23)\\
&= \mathbb{P}\left[\frac{p_i}{p_w}e^{-x} \le -\log h_i, \forall i\right] && (24)\\
&= \prod_{i=1}^{|\mathcal{V}|} \mathbb{P}\left[\frac{p_i}{p_w}e^{-x} \le -\log h_i\right] && (25)\\
&= \prod_{i=1}^{|\mathcal{V}|} 1 - \mathbb{P}\left[-\log h_i \le \frac{p_i}{p_w}e^{-x}\right] && (26)\\
&= \prod_{i=1}^{|\mathcal{V}|} 1 - (1 - e^{-\frac{p_i}{p_w}e^{-x}}) && (27)\\
&= \prod_{i=1}^{|\mathcal{V}|} e^{-\frac{p_i}{p_w}e^{-x}} && (28)\\
&= e^{\sum_{i=1}^{|\mathcal{V}|} \frac{-p_i}{p_w}e^{-x}} && (29)\\
&= e^{\frac{-e^{-x}}{p_w}} && (30)
\end{aligned}
$$

In equation (22), we utilize the fact that $p_i \propto e^{l_i}$ and $\xi_i = -\log(-\log h_i)$. Equation (24) leverages the independence of h_i. Equation (26) uses the fact that $-logh_i \sim \text{Exp}(1)$. Finally, equation (29) employs the fact that $\sum_{i=1}^{|\mathcal{V}|} p_i = 1$. Hence, the density function:

$$
f(x) = F^{'}(x) = \frac{e^{-x}}{p_w} e^{\frac{-e^{-x}}{p_w}}
$$

The expectation:

$$\begin{aligned}\mathbb{E}[\xi_t[w_t]] &= \mathbb{E}[X] &(31)\\ &= \int_{-\infty}^{+\infty} x f(x) dx &(32)\\ &= -\int_{-\infty}^{+\infty} x \frac{e^x}{p_w} e^{\frac{-e^x}{p_w}} dx &(33)\\ &= -\frac{1}{p_w}[p_w \log p_w - \gamma p_w] &(34)\\ &= -\log p_w + \gamma &(35)\end{aligned}$$

The equation (32) use the fact that:

$$\int_{-\infty}^{+\infty} x e^x e^{\frac{-e^x}{t}} dx = t \log t - \gamma t$$

We now prove the fact. This is not a standard integral that can be solved by elementary functions. However, we can attempt to solve it using the substitution method and some properties of the Gamma and incomplete Gamma functions, which are commonly used to handle integrals involving exponentials of exponentials.

$$\begin{aligned}&\int_{-\infty}^{+\infty} x e^x e^{\frac{-e^x}{t}} dx &(36)\\ &= \int_{0}^{+\infty} \log(u) e^{-u/t} du &(37)\\ &= \int_{0}^{+\infty} \log(vt) e^{-v} t dv &(38)\\ &= t \int_{0}^{+\infty} (\log(v) + \log(t)) e^{-v} dv &(39)\\ &= t \log(t) \int_{0}^{+\infty} e^{-v} dv + t \int_{0}^{+\infty} \log(v) e^{-v} dv &(40)\\ &= t \log(t) + t \int_{0}^{+\infty} \log(v) e^{-v} dv &(41)\\ &= t \log(t) - \gamma t &(42)\end{aligned}$$

In Equation(35), we use variable substitution $u = e^x$, In Equation(36), we use variable substitution $v = \frac{u}{t}$, In Equation (39), we use the definition of Euler-Mascheroni constant: $\gamma = -\int_0^{+\infty} \log(v) e^{-v} dv$.

As for the variance of the per-token score for watermarked text, we can also derive it via the probability density function $f(x)$.

$$\begin{aligned}&\text{Var}[s_t] &(43)\\ &= \text{Var}[\xi_t[w_t]] &(44)\\ &= \mathbb{E}[X^2] - (\mathbb{E}[X])^2 &(45)\\ &= \int_{-\infty}^{+\infty} x^2 f(x) dx - (-\log p_w + \gamma)^2 &(46)\\ &= \int_{-\infty}^{+\infty} x^2 \frac{e^{-x}}{p_w} e^{\frac{-e^{-x}}{p_w}} dx - (-\log p_w + \gamma)^2 &(47)\\ &\leq \frac{2{p_w}^2}{(1-p_w)^3} + \frac{2}{p_w} - (-\log p_w + \gamma)^2 &(48)\end{aligned}$$

In equation(48), we use the fact that

$$\int_{-\infty}^{+\infty} x^2 \frac{e^{-x}}{p_w} e^{\frac{-e^{-x}}{p_w}} dx \leq \frac{2{p_w}^2}{(1-p_w)^3} + \frac{2}{p_w}$$

We now prove it:

$$\begin{aligned}&\int_{0}^{+\infty} x^2 \frac{e^{-x}}{p_w} e^{\frac{-e^{-x}}{p_w}} dx &(49)\\ &\leq \int_{0}^{+\infty} x^2 \frac{e^{-x}}{p_w} dx &(50)\\ &= \frac{2}{p_w} &(51)\end{aligned}$$

$$\begin{aligned}&\int_{-\infty}^{0} x^2 \frac{e^{-x}}{p_w} e^{\frac{-e^{-x}}{p_w}} dx &(52)\\ &= \int_{0}^{+\infty} x^2 \frac{e^{x}}{p_w} e^{\frac{-e^{x}}{p_w}} dx &(53)\\ &= \int_{0}^{+\infty} \frac{x^2}{p_w} e^{(x - \frac{e^x}{p_w})} dx &(54)\\ &\leq \int_{0}^{+\infty} \frac{x^2}{p_w} e^{(1-\frac{1}{p_w})x} dx &(55)\\ &= \frac{2{p_w}^2}{(1-p_w)^3} &(56)\end{aligned}$$

A similar theorem also holds for the Exponential watermark. For unwatermarked text:

$$\mathbb{E}[s_t] = \mathbb{E}[-\log(1 - \xi_t[w_t])] = 1$$
$$\text{Var}[s_t] = \text{Var}[-\log(1 - \xi_t[w_t])] = 1$$

For watermarked text,

$$\begin{aligned}\mathbb{E}[s_t] &= \mathbb{E}[-\log(1 - \xi_t[w_t])]\\ &\geq 1 + (\frac{\pi^2}{6} - 1)(-p_w \log p_w),\\ \text{Var}[s_t] &= \text{Var}[-\log(1 - \xi_t[w_t])]\\ &= \psi_1(1) - \psi_1(1 + \frac{1}{p_w}),\end{aligned}$$

where ψ_1 is the trigamma function. The proof can be found in Fernandez et al. (2023)

		Diversity			Detectability			Quality
		Self-Bleu ↓	Dist-1 ↑	Dist-2 ↑	AUROC ↑	FPR ↓	FNR ↓	PPL ↓
Exponential	vanilla	1.000	0.010	0.014	1.000	0.000	**0.000**	**10.953**
	drop_prob=0.05	0.367	0.222	0.529	1.000	0.000	**0.000**	11.450
	drop_prob=0.10	0.227	0.254	0.633	1.000	0.000	0.001	11.423
	drop_prob=0.20	0.146	0.300	0.733	1.000	0.000	0.001	11.839
	drop_prob=0.30	0.113	0.307	0.766	1.000	0.000	0.002	11.911
	drop_prob=0.40	**0.087**	0.317	0.788	1.000	0.001	0.005	11.964
	shift_max=10	0.991	0.079	0.146	0.999	0.000	0.003	11.307
	shift_max=30	0.798	0.158	0.333	0.999	0.000	0.002	11.084
	shift_max=50	0.645	0.184	0.403	1.000	0.000	0.003	11.222
	shift_max=100	0.414	0.221	0.496	0.999	0.000	0.003	11.102
	shift_max=200	0.247	0.235	0.546	0.999	0.000	0.004	11.068
	soft_temp=0.1	0.388	0.210	0.490	1.000	0.000	**0.000**	11.218
	soft_temp=0.2	0.244	0.238	0.565	1.000	0.000	0.001	11.353
	soft_temp=0.3	0.202	0.265	0.630	1.000	0.000	0.001	11.610
	soft_temp=0.4	0.169	0.275	0.669	1.000	0.000	0.001	11.874
	soft_temp=0.5	0.146	0.285	0.686	1.000	0.000	0.001	12.222
Logits-Addition	vanilla	1.000	0.010	0.014	1.000	0.000	0.000	10.953
	drop_prob=0.05	0.421	0.205	0.493	0.999	0.000	0.003	11.561
	drop_prob=0.10	0.209	0.268	0.652	0.999	0.000	0.003	11.754
	drop_prob=0.20	0.143	0.292	0.729	0.999	0.000	0.005	11.997
	drop_prob=0.30	0.097	0.309	0.774	0.999	0.001	0.005	11.890
	drop_prob=0.40	0.093	**0.319**	**0.790**	0.999	0.003	0.006	12.156
	shift_max=10	0.991	0.078	0.144	0.999	0.000	0.006	11.228
	shift_max=30	0.806	0.159	0.335	0.998	0.002	0.006	11.243
	shift_max=50	0.627	0.188	0.412	0.998	0.000	0.006	11.250
	shift_max=100	0.417	0.220	0.497	0.998	0.000	0.006	11.536
	shift_max=200	0.246	0.242	0.559	0.998	0.001	0.007	11.243
	soft_temp=0.1	0.370	0.213	0.497	1.000	0.000	0.001	11.159
	soft_temp=0.2	0.227	0.243	0.581	1.000	0.000	0.002	11.309
	soft_temp=0.3	0.158	0.254	0.608	1.000	0.000	0.001	11.820
	soft_temp=0.4	0.121	0.276	0.661	1.000	0.000	0.001	12.831
	soft_temp=0.5	0.100	0.298	0.699	1.000	0.000	0.001	14.140

Table 4: Comparison of three variants of both Exponential and Logits-Addition watermarks in the Completion task. The variants include **drop_prob=0.2**, sampling from the language model directly at a 0.2 probability; **shift_max=100**, where the watermark key is cyclically shifted within a 0-100 range; and **soft_temp=0.3**, which uses a softmax sampling with a temperature of 0.3 to balance randomness. **Vanilla** is the original two GumbelMax watermarks without any technique to enhance diversity. The detectability is measured by 100 detection tokens. Note that Logits-Addition+soft_temp is our GumbelSoft watermark.

C Experiment details

We describe all experiment details here. We run all experiments five times and report the average value. The variance of the five repeated experiments is negligible, approaching zero; therefore, it has been excluded from the table. The detailed comparison of our GumbelSoft watermark with other watermark variants in the Text Completion task is presented in Table 4.

C.1 Diversity

We began by carefully selecting 40 high-entropy prompts to elicit a wide range of completions. These prompts were split evenly into two categories: 20 prompts followed a Completion format tailored for Llama2-7b, while the remaining 20 were structured in a QA format, specifically designed for Llama2-7b-chat. Each prompt was queried 50 times, and we assessed the resulting completions using metrics such as Self-Bleu, Distinct 1-gram, and Distinct 2-gram. The average values of these metrics were then computed for the 20 prompts in each category. We control the max generation length for each prompt to be 256 tokens.

For the soft_temp parameter, we tested five different temperature settings: 0.1, 0.2, 0.3, 0.4, and 0.5. In the case of the shifted watermark key, we experimented with five maximum shift values: 10, 30, 50, 100, and 200. For drop probability, the tested probabilities were 5%, 10%, 15%, 20%, and

40%. We evaluated detectability and quality using a sample of 100 generated tokens, while diversity assessments were conducted with a sample size of 256 tokens.

C.2 Detectability

The objective of text watermarking is to embed a concealed pattern into generated texts and subsequently detect this pattern to ascertain if the text is watermarked. We gathered 1,000 lengthy texts from the news-like validation subset of the C4 dataset, dividing each text into two parts: the first 50 words as prompt and the remaining as gold-completion. For each watermarking scheme, we utilized Llama2-7b to create both watermarked and unwatermarked completions for these 1,000 prompts. The effectiveness of each scheme was then assessed using the corresponding detector to evaluate 2,000 completions. Key metrics reported include AUROC (Area Under the Receiver Operating Characteristic), FPR (False Positive Rate) at a fixed FNR (False Negative Rate) of 0.01, and FNR at a fixed FPR of 0.01.

Additionally, we compiled 1,000 lengthy texts from the alpaca dataset. Unlike the C4 dataset, here we used only the question as a prompt to query Llama2-7b-chat, with the subsequent detection process mirroring that of the C4 dataset.

In line with the detectability theorem by Chakraborty et al. (2023), we anticipate higher detectability in longer texts. Therefore, we report detection metrics for generated token lengths of 40, 60, 80, and 100. However, for quality assessment, we calculate perplexity only for texts with 100 generated tokens, as fewer tokens would inadequately represent quality measures. We use llama2-13b and llama2-13b-chat to evaluate ppl for the texts generated by llama2-7b and llama2-7b-chat respectively.

The hyper-parameters employed for each watermarking scheme are specified as follows: All experiments are conducted at a temperature setting of 1, except the GumbelSoft, which utilized a temperature setting of 0.3 to achieve an equilibrium between detectability and generation diversity. For KGW, we adopt $\delta = 2$ and $\gamma = 0.1$, following the recommendations by Kirchenbauer et al. (2023). For Dipmark, the parameters are set to $\alpha = 0.45$ and $\gamma = 0.5$, by Wu et al. (2023b). Regarding ITS, we utilize a sample of 500 texts from the C4 subset for the computation of reference scores. We repeat the experiment 5 times to calculate the average value for each metric.

C.3 Robustness

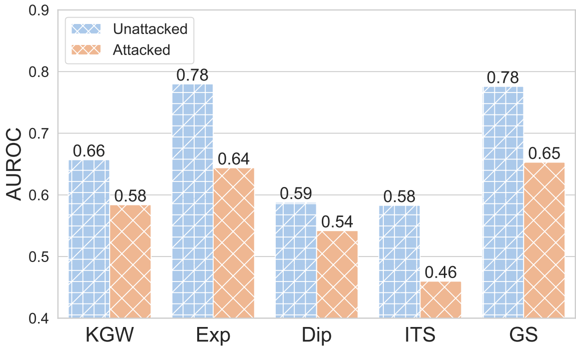

Figure 7: Comparison of robustness of decoding-based watermark on QA task. Blue histograms indicate unattacked conditions and red histograms show attacked scenarios. The AUROC is calculated for 40 detection tokens, with GumbelSoft set at a 0.3 temperature. **Exp**, **Dip**, and **GS** refer to Exponential, Dipmark, and GumbelSoft, respectively.

T5-span Attack We employ the T5-span attack (Kirchenbauer et al., 2023) on both watermarked and unwatermarked texts. Each word in a text undergoes a potential attack with a probability of 0.5. For attacked words, we use their immediate five-word context (preceding and following) and apply t5-large (Raffel et al., 2019) for context-based word prediction, replacing the original word with the predicted one. This process may occasionally retain the original word; however, we opt not to enforce unique substitutions to avoid excessive time consumption.

Paraphrase Attack We employ GPT-3.5 Turbo to perform paraphrase attacks on the completion task. The results, presented in Table 5, indicate that the paraphrase attack is significantly more potent than the T5-span attack due to its substantial modifications to the context. Consequently, all decoding-based watermarks exhibit poor performance under such an attack.

C.4 Downstream Task Evaluation

We further evaluate all decoding-based watermarks on downstream tasks to compare their generation quality.

Experiment Setting. We conducted quality assessment experiments on two sequence-to-sequence (seq2seq) downstream tasks: constrained text generation and summarization. In the constrained text generation task, the model receives

		# tokens=40			# tokens=60			# tokens=100		
		AUROC ↑	FPR ↓	FNR ↓	AUROC ↑	FPR ↓	FNR ↓	AUROC ↑	FPR ↓	FNR ↓
Completion	KGW	0.580	0.989	0.981	0.598	0.990	0.971	0.632	0.979	0.957
	Exponential	**0.689**	0.975	**0.857**	0.721	0.969	**0.819**	0.768	0.951	**0.775**
	Dipmark	0.553	0.984	0.986	0.570	0.985	0.983	0.589	0.985	0.976
	ITS	0.592	1.000	1.000	0.609	1.000	1.000	0.649	1.000	1.000
	GumbelSoft	0.685	**0.971**	0.862	**0.723**	**0.966**	0.841	**0.770**	**0.951**	0.781

Table 5: A comparative analysis of the detectability across various decoding-based watermarking schemes under GPT-3.5 Turbo Paraphrase attack. The temperature for GumbelSoft is set to 0.3. All decoding-based watermarks perform poorly due to their reliance on previous contexts.

Methods	Average Cover Rate↑	Variance Cover Rate↓	Average PPL↓
Unwatermarked	69.091	480.622	4.824
KGW	70.015	368.197	5.695
Exponential	70.396	264.503	4.919
Dipmark	69.308	436.202	4.115
ITS	70.137	382.446	4.855
GumbelSoft	68.681	328.752	5.103

Table 6: Comparison of various watermarks on the constrained text generation task reveals that all watermarks perform similarly.

Model	Average SBERT↑	Variance SBERT↓	Average Rouge1↑	Variance Rouge1↓
Unwatermarked	0.619	0.036	0.219	0.006
KGW	0.637	0.029	0.225	0.006
Exponential	0.646	0.019	0.223	0.006
Dipmark	0.625	0.037	0.219	0.007
ITS	0.575	0.058	0.204	0.010
GumbelSoft	0.650	0.020	0.223	0.006

Table 7: Comparison of various watermarks on the summarization task indicates that the GumbelSoft watermark performs slightly better than the others.

a list of words and is tasked with generating a coherent sentence incorporating these words. In the summarization task, the model is given a long article (approximately 1500 tokens) and is expected to generate a concise summary. We utilized the Llama2-7b-chat model as our test model. For the constrained text generation and summarization tasks, we used the hard subset of CommonGen and CNN-DM datasets, respectively, each comprising 200 examples.

Metric and Prompt. For the constrained text generation task, we employed cover rate as the metric, which measures the percentage of provided words included in the generated sentence. The instruction prompt used was: “Use the following words to create a sentence. The sentence should contain all provided words.” For the summarization task, we utilized the F-measure of ROUGE-1 and cosine similarity of Sentence-BERT ’all-MiniLM-L6-v2’ as metrics. The instruction prompt used was: "Please help me summarize the given article."

Results. The results are presented in Table 6 and Table 7. The data indicate that our Gumbelsoft model outperforms other decoding-based watermarks in terms of sentence-BERT cosine similarity for the summarization task. In the constrained text generation task, while all methods exhibit comparable performance, the extremely high variance diminishes the significance of the analysis.

D Responsible NLP Research.

The C4 dataset is under the terms of ODC-BY and the Alpaca dataset is under the terms of Creative Commons NonCommercial (CC BY-NC 4.0). Our research fully obeys these licenses. C4 and Alpaca datasets are publicly available and do not contain private information for any individual.