Enhancing LLM Watermark Resilience Against Both Scrubbing and Spoofing Attacks

Huanming Shen[1,2]* huanmingshen@std.uestc.edu.cn
Baizhou Huang[1]* hbz19@pku.cn
Xiaojun Wan[1]† wanxiaojun@pku.cn

[1]Wangxuan Institute of Computer Technology, Peking University
[2]University of Electronic Science and Technology of China

Abstract

Watermarking is widely regarded as a promising defense against the misuse of large language models (LLMs); however, existing methods are fundamentally constrained by their vulnerability to scrubbing and spoofing attacks. This vulnerability stems from an inherent trade-off governed by watermark window size: smaller windows resist scrubbing better but are easier to reverse-engineer, enabling low-cost statistics-based spoofing attacks. This work expands the trade-off boundary by introducing a novel mechanism, equivalent texture keys, where multiple tokens within a watermark window can independently support the detection. Based on the redundancy, we propose a watermark scheme with **S**ub-vocabulary decomposed **E**quivalent t**E**xture **K**ey (**SEEK**). SEEK achieves a Pareto improvement, enhancing robustness to scrubbing attacks without sacrificing resistance to spoofing.

1 Introduction

Recent advances in large language models [12, 8, 52, 59, 60, 44, 45] have significantly improved the realism of AI-generated text, making it increasingly difficult to distinguish from human-written content. However, this progress also raises concerns about potential misuse, including disinformation dissemination [64], automated phishing [57], and academic integrity [50]. In response, detecting LLM-generated text has become an active area of research, with growing interest in techniques for enhancing provenance tracking and content verification.

One proactive approach involves watermarking, wherein the LLM's output distribution is subtly altered to embed an imperceptible signature within the generated text [29, 32, 13, 72]. This signature can subsequently be verified by a detector equipped with a secret key held by the LLM provider. Compared to post-hoc detection methods [15, 43, 24, 68], watermarking offers notable advantages, including the preservation of output quality and a provably low false positive rate (below 10^{-3}) [19]. However, watermarking remains vulnerable to two types of adversarial attacks: 1) **scrubbing attacks** which generate a semantically equivalent paraphrase to disturb watermark patterns, thereby bypassing the watermark detection and 2) **spoofing attacks** which mimic the watermark patterns to inject target watermarks into harmful text, fabricating content that appears to be generated by the victim LLM.

A widely acknowledged perspective in existing research [27, 30, 34, 46, 35] is that distribution-modifying watermarking exhibits an inherent trade-off between scrubbing robustness and spoofing robustness. Among such methods, KGW [29] stands out as a representative approach, introducing a family of h-gram statistical watermarks [22, 72, 30]. They modify the language model's next token distribution at each step based on the preceding h tokens, referred to as the watermark window.

*Equal contribution
†Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

A larger watermark window requires longer unaltered text segments for reliable detection, thus increasing susceptibility to localized edits, i.e., scrubbing. Therefore, to maintain resilience against scrubbing attacks like paraphrasing, these methods typically restrict the watermark window size to fewer than four tokens. However, the small window setting also introduces a structural limitation that can be exploited by spoofing attacks. Recent study [27] has revealed that by collecting moderate samples from the watermarked LLM, adversaries can statistically learn the mapping from watermark windows to corresponding watermarked outputs. This enables the generation of high-quality spoofed text that closely mimics the characteristics of genuinely watermarked text. In essence, the efficacy of such adversarial attacks is not primarily due to their sophistication, but rather the inherent limitations of conventional schemes that force a direct compromise between scrubbing and spoofing robustness via the singular control of the watermark window size, as shown in Figure 1. This presents a compelling research question:

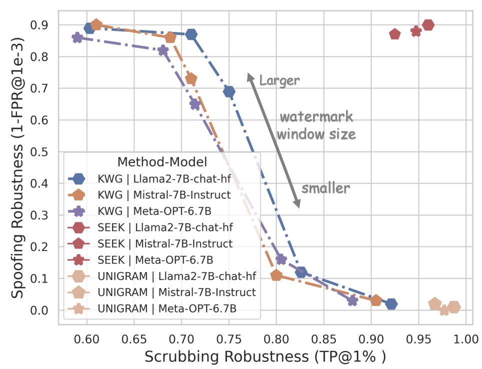

Figure 1: Performance of different schemes under scrubbing and spoofing attacks. Varying the watermark window size induces a trade-off between scrubbing and spoofing robustness. Scrubbing robustness is evaluated using DIPPER [31] on the C4-RealNewsLike dataset. Spoofing robustness is evaluated using statistics-based attacks [27] on the Dolly-CW dataset. Ours achieves improved robustness on both axes, reaching Pareto optimality.

How can watermarking schemes be enhanced to simultaneously ensure robustness against scrubbing and resistance to spoofing attacks?

To address this question, we investigate an overlooked factor influencing robustness against scrubbing attacks beyond the watermark window size. We identify a phenomenon of equivalent texture keys, where multiple tokens within a watermark window can independently support the detection of the watermark pattern at the subsequent timestep. This redundancy enables a path towards Pareto improvements: by increasing the number of equivalent texture keys, scrubbing robustness can be significantly enhanced even when employing large watermark windows, without compromising resistance to spoofing attacks.

Building on this insight, our initial approach increases the prevalence of equivalent texture keys by reducing the output space of the hash function typically used in watermarking. However, recognizing that a smaller hash output space could potentially degrade text quality, we introduce our final watermark scheme, **SEEK** (**S**ub-vocabulary decomposed **E**quivalent t**E**xture **K**ey). SEEK ingeniously mitigates this quality concern by decoupling the watermark construction process across disjoint sub-vocabularies. This allows for a high density of equivalent texture keys, bolstering robustness, while simultaneously preserving the quality of the generated text. Our contributions are as follows:

- We reveal that equivalent texture keys, where multiple tokens within a watermark window independently support detection, are critical for enhancing scrubbing robustness under large windows.
- We propose a novel scheme that improves robustness by leveraging the equivalent texture key mechanism, while simultaneously guaranteeing text quality through the decomposition of watermark construction across disjoint sub-vocabularies.
- The experiments demonstrate that SEEK outperforms KGW-family baselines, achieving substantial improvements in scrubbing and spoofing robustness across datasets, establishing a new Pareto frontier, yielding spoofing robustness gains of +88.2%/+92.3%/+82.0% on Dolly-CW/MMW-BookReports/MMW-FakeNews and scrubbing robustness gains of +10.2%/+13.4%/+8.6% on WikiText/C4/LFQA compared to KGW-Min with 4-gram watermark window.

2 Related work

LLM Watermarking technology. Watermarking natural language faces challenges due to the discrete and symbolic nature of text [56]. Early methods employed rule-based techniques such as syntactic restructuring [4], synonym substitution [58], and paraphrasing [5]. Foundational pioneering work in LLM watermarking includes KGW [29, 30] and AAR [2] builds watermarks on statistics for a low and provable false positive rate. Specifically, watermark detection was framed as a hypothesis testing problem, where the alternative hypothesis models text generation from a modified distribution. The KGW method partitions the vocabulary into a "green" list and a "red" list to modify the next token's distribution, which significantly influenced numerous follow-up watermark works [32, 40, 33, 72, 34, 39]. Our work focuses on the security properties of KGW-family watermarking schemes.

Scrubbing attacks. Scrubbing attacks fundamentally stem from text editing operations. Early studies [72, 32, 29] evaluated the robustness of watermarking schemes using simple perturbations such as random word deletion or substitution. In addition, some work [30, 72] has explored manual paraphrasing by human annotators to assess watermark removal under high-quality rewriting. Another line of work aims to develop black-box scrubbing attacks, like Zhang et al. [69] and Sadasivan et al. [54] propose recursive paraphrasing pipelines, while Huang et al. [25] formulate the scrubbing task as a constrained optimization problem. Krishna et al. [31] propose a systematic approach to evade AI-generated text detection by training a paraphrasing model.

Spoofing attacks. Early work [54, 22, 16, 18] has demonstrated proof-of-concept spoofing against simple $h = 1$ watermark schemes known as KGW-Left [29]. Gu et al. [18] propose a distillation-based spoofing method, which requires considerable training overhead. Zhang et al. [71] formalize spoofing as a mixed-integer programming problem; however, their method depends on access to the model's full vocabulary and tokenizer. In contrast, Jovanović et al. [27] proposes a highly effective statistics-based spoofing attack that approximates the watermark distribution via token frequency analysis, without any model-specific access. By learning watermark prefix $p_w(x_n|x_{n-h:n-1})$ and non-watermark prefix $p_m(x_n|x_{n-h:n-1})$, they successfully spoof KGW-Min [30] for under \$50.

3 Preliminaries

3.1 Watermarking in LLMs

Large Language models. Following the notation from prior work [66, 23, 25, 40, 32], we define the generation process of a LLM. Let $\mathcal{M}$ denote a pretrained LLM with a vocabulary $\mathcal{V}$, typically comprising over 50,000 tokens (i.e., $|\mathcal{V}| > 50{,}000$) [32]. Given an input $x_{1:n-1} = \{x_1, x_2, \ldots, x_{n-1}\}$, the model generates a sequence by computing token-wise logit scores ℓ_n and sampling tokens from the output distribution $x_n \sim P_{\mathcal{M}}(\cdot \mid x_{1:n-1})$ in an autoregressive fashion.

LLM watermarking. Watermarking alters the origin model's generation distribution $P_M(\cdot|x_{1:n-1})$, yielding a modified distribution $P_W(\cdot|x_{1:n-1}) = \mathcal{T}(P_M(\cdot|x_{1:n-1}), \theta_n)$, where θ_n denotes the cipher for the current step. KGW-family methods typically conduct the modification $\mathcal{T}$ through a random partition of the vocabulary seeded by the cipher θ_n. Specifically, a subset $G \subseteq \mathcal{V}$ of predefined size $\gamma|\mathcal{V}|$ is sampled as the green list, while the rest is termed as the red list $R = \mathcal{V} \setminus G$. Then a positive bias δ is added to the logits of tokens in the green list, shifting the distribution to favor sampling green list tokens. Specifically, the modified logit of token t, denoted by $\hat{\ell}_n[t]$, is formulated as:

$$\hat{\ell}_n[t] = \ell_n[t] + \delta \cdot \mathbf{1}_G(t) \tag{1}$$

where $\mathbf{1}(\cdot)$ is the indicator function. During the detection phase, A one-sided significance test is conducted to distinguish between watermarked and non-watermarked text [29, 25, 32]. Formally, given a candidate sequence $\tilde{x}_{1:N} = \{x_1, \ldots, x_N\}$, the detection problem is framed as a z-statistic test with tokens in all timesteps as i.i.d. samples. We reconstruct the green list at each step n using the cipher θ_n, and count the green tokens in the candidate sequence as the test statistic, from which the z-score is computed. If z-score exceeds a certain threshold, we conclude that the watermark is embedded in x and thus generated by the LLM $\mathcal{M}$. Notably, this test fully depends on the cipher θ_n in the candidate, rendering watermark detection infeasible if it is disturbed. Kirchenbauer et al. [30] proposed a technique called self-seeding to incorporate the next token x_n as part of the watermark window, which extends the watermark window by one token. The self-seeding mechanism and the watermark detection process are described in detail in Appendix C.1 and Appendix C.3, respectively.

Derivation of the cipher θ_n. The cipher $\theta_n = \xi \cdot \zeta$ consists of two components: a *secret key* ξ held by the service provider and a *texture key* [66] ζ derived from the watermark window. Specifically, given a hash function H with a hash space of $\{1, 2, 3, ..., d\}$, a hash signature $I = \{H(x_{n-k})|1 \leq k \leq h\}$ of current watermark window is firstly calculated. Then an aggregation function $f : \mathbb{N}^h \to \mathbb{N}$ is applied on the signature to generate the texture key ζ. Among several variants of the aggregation function, the f =*Min* variant exhibits superior robustness [16, 37], which is formulated as:

$$\theta_n = \zeta \cdot \xi = f(I) \cdot \xi = \text{Min}\, I \cdot \xi = \text{Min}\left\{H(x_{n-h}) \cdot \xi, \ldots, H(x_{n-1}) \cdot \xi)\right\} \tag{2}$$

We also present other aggregation function f variants in detail in Appendix C.2.

3.2 Threat Models

Following previous works [27, 32, 48, 25], the threat models involve two primary entities: *victim* and *attacker*. The victim operates a special language model $\mathcal{M}$ protected by watermarking and provides services externally via an API. The attacker aims to compromise the watermark detection algorithm through either scrubbing attacks or spoofing attacks. Under standard security assumptions (i.e., Kerckhoffs's principle [28]), we follow [65, 27] to assume the attacker has complete knowledge of the watermarking scheme, and can query the victim model via an API interface.

Scrubbing's objective. Scrubbing attacks refer to adversarial attempts to remove watermarks from protected text. Formally, given a watermarked response, an attacker aims at producing a semantically equivalent paraphrased output classified as non-watermarked text by the victim's detector.

Spoofing's objective. The objective of a spoofing attack is to construct a watermarking model $\mathcal{M}'$ that mimics the text distribution of the victim model $\mathcal{M}$. Subsequently, the attacker is able to generate a malicious message that is classified as watermarked text by the victim's detector.

4 Method

4.1 Motivation: Pareto Improvements via Hash Collisions

A robust watermarking algorithm must not only accurately distinguish watermarked text but also maintain detection reliability under various adversarial perturbations. However, a fundamental trade-off exists between robustness against scrubbing attacks and resistance to spoofing attacks as shown in Figure 1, which poses a significant challenge to the practical deployment of watermarking methods.

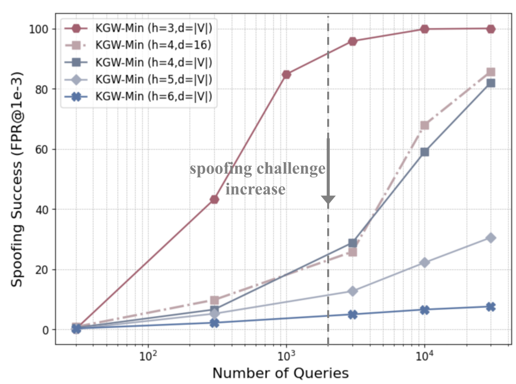

Figure 2: Performance of different watermark window schemes under spoofing attack. The attack is conducted using 500 malicious texts generated by Dolly-CW, targeting a calibrated detector under FPR of 0.1%.

Table 1: Comparison of the performance of various watermarking schemes against the ChatGPT scrubbing attack on the C4-Realnewslike dataset. h represents the watermark window size, and d indicates the cardinality of the hash space.

Schemes	TP@1% ↑	TP@5% ↑
Origin		
KGW-Min (h=4, d=$\lvert\mathcal{V}\rvert$)	73.5	86.5
KGW-Min (h=8, d=$\lvert\mathcal{V}\rvert$)	49.5	70.7
KGW-Min (h=16, d=$\lvert\mathcal{V}\rvert$)	43.1	58.8
KGW-Min (h=32, d=$\lvert\mathcal{V}\rvert$)	32.4	47.6
With Equivalent Texture Keys		
UNIGRAM (h=0, d=1)	91.8	94.9
KGW-Min (h=4, d=16)	80.3	90.2
KGW-Min (h=8, d=16)	78.9	87.8
KGW-Min (h=16, d=16)	72.9	85.9
KGW-Min (h=32, d=16)	53.4	74.6

Finding I: *Statistical spoofing attacks suffer from sparsity: as the watermark window increases, the sample complexity grows exponentially, thereby reducing spoofing success rates.*

The efficacy of statistics-based spoofing [27] stems from the statistical analysis of the watermark window, enabling the attacker to estimate the mapping between watermark windows and corresponding watermarked tokens. Intuitively, the complexity of such attacks is determined by the diversity of watermark windows. As shown in Figure 1 and 2, increasing the size of the watermark window leads to a substantial decline in spoofing success rates. This observation aligns with the sparsity challenge highlighted in [27], where sample complexity in statistical spoofing attacks scales as $\mathcal{O}(|\mathcal{V}|^h)$.

Finding II: *For watermarking schemes with the same window size, reducing the cardinality of the hash space improves robustness against scrubbing attacks without compromising spoofing resistance.*

We investigate a variant of KGW that operates over a reduced hash space, thereby increasing the likelihood of hash collisions[3] when deriving texture keys in Eq 2. As shown in Table 1, this variant achieves markedly gains in robustness against scrubbing attacks while maintaining watermark window size. We attribute the improvement to the increased frequency of hash collisions in the watermark window, which enables multiple tokens to independently reconstruct texture key ζ for valid detection.

Consider KGW-MIN as an example, where texture key $\zeta = \mathrm{Min}\,\{H(x_{n-h}), \ldots, H(x_{n-1}))\}$. In the presence of frequent hash collisions, multiple tokens x_j in the watermark window may satisfy $H(x_j) = \mathrm{Min}\, I$, enabling each to independently support watermark detection. We refer to this phenomenon as the emergence of *Equivalent Texture Keys*. This redundancy enhances robustness: an attacker now needs to disturb all equivalent texture key tokens to remove the watermark. The challenge of scrubbing attacks towards our proposed variant can be formulated as follows:

Proposition 4.1. *Given a hash function with space dimension d and a watermark window size h, the probability of a hash collision occurring can be approximated by $p(h, d) \geq 1 - e^{\frac{-h(h-1)}{2d}}$.*

Proposition 4.2. *The expected number of tokens that must be erased to eliminate the watermark is expressed as $\sum_{m=1}^{d} \frac{h}{d} \left(\frac{d-m+1}{d}\right)^{h-1}$, which is monotonically non-increasing in d (strictly decreasing for $h \geq 2$).*

Formal proofs and corresponding visualizations for these propositions are presented in Appendix C.4. From this perspective, UNIGRAM [53], a scheme known for its exceptional resilience to scrubbing, can also be interpreted as a special case of our proposed variant with the minimal hash space of cardinality $d = 1$. On the other hand, reducing the hash space size does not influence the watermark window size, thereby maintaining similar spoofing resistance[4]. Experimental results in Figure 2 indicate that the success rate of spoofing attacks is not affected by d.

Taken together, these two findings underscore a key insight: the introduction of equivalent texture keys yields a form of Pareto improvement, enhancing robustness against scrubbing without sacrificing spoofing resistance, offering a promising pathway toward more practical watermark designs.

4.2 SEEK: Sub-vocabulary Decomposed Equivalent Texture Keys

While reducing the size of the hash space yields significant improvements in watermark robustness, it inadvertently restricts the diversity of green-red list partitions. In the KGW-Min scheme, the diversity is upper bounded by $\mathcal{O}(d)$, which severely degrades the text quality. As illustrated in Figure 3, we observe a clear degradation in generation quality with the decrease of cardinality d. Prior research [47, 35] has also raised concerns that reducing the number of possible partitions may increase the risk of underlying pattern learning. In this section, we seek to harness the potential of equivalent texture keys while mitigating their adverse impact on generation quality.

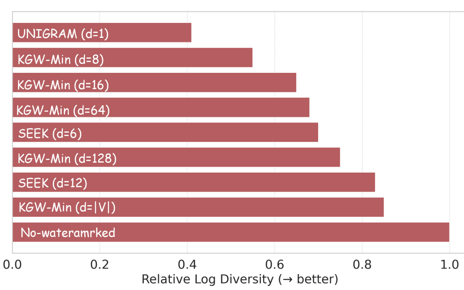

Figure 3: An analysis of generation quality across different schemes on a subset of C4 by Log Diversity metric, relative to the unwatermarked text.

We first revisit the variant proposed in the previous section. Given the hash function $H(\cdot)$ maps to $\{1, 2, ..., d\}$, the cipher derivation in Equation 2 can be reformulated as follows:

$$\forall i \in \{1, 2, ..., d\},\, \theta_n^i = i \cdot \xi \qquad \theta_n = \mathrm{Min}\{\theta_n^i | i \in I\} \tag{3}$$

Since each θ_n^i corresponds to one unique green list partition of the vocabulary, $\mathrm{Min}\{\theta_n^i | i \in I\}$ essentially conducts a selection over all possible partitions, which is thereby upper bounded by the hash space size d. This formulation reveals that the reason for the inferior diversity of green-red list partitions stems from the dependency on one single cipher.

To overcome this, we propose a watermark scheme with **S**ub-vocabulary decomposed **E**quivalent t**E**xture **K**eys (**SEEK**), which retains robustness via equivalent texture keys but enables $\mathcal{O}(2^d)$ distinct green-red partitions. The main idea is to distribute green list construction across multiple

[3] A hash collision occurs when two tokens in a watermark window yield same hash value under $H(\cdot)$.

[4] However, if d is reduced to 1, KGW-Min degenerates into a UNIGRAM scheme.

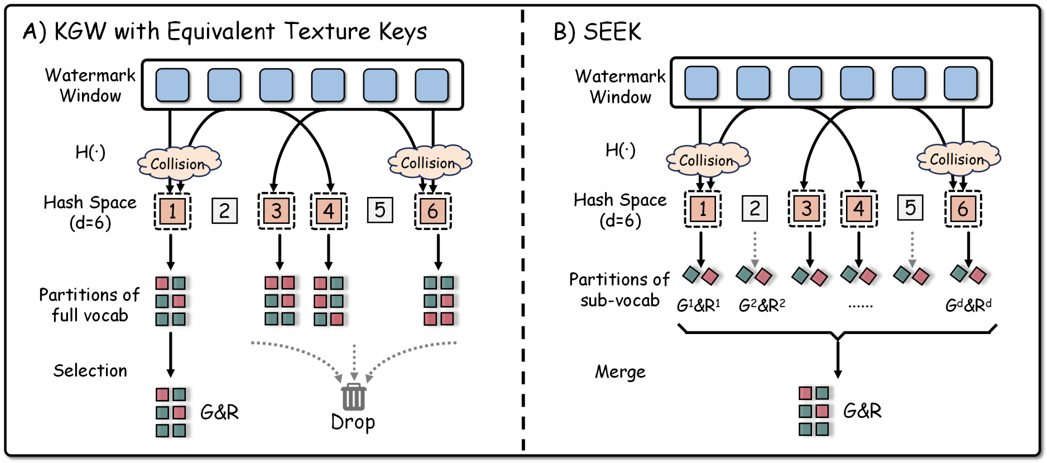

Figure 4: (A) KGW-MIN with equivalent texture keys proposed in Section 4.1. Each value in hash space derives a distinct θ^i to generate a partition of vocabulary. We then select one as the final green list. (B) SEEK proposed in Section 4.2. Different from (A), each value in the hash space only contributes to a partition of a sub-vocabulary G^i. We then merge all partitions as the final green list.

sub-vocabularies to enhance diversity, while preserving frequent equivalent texture keys within each, as illustrated in Figure 4.

Specifically, we split the vocabulary $\mathcal{V}$ into d sub-vocabularies $\{v_1, v_2, \ldots, v_d\}$, and independently assign each with a cipher θ_n^i to generate an individual sub-green list G^i. The final green list G is then constructed by unifying all sub-green lists, and the complementary set forms the red list R. To induce equivalent texture keys within each sub-vocabulary, we derive the cipher θ_n^i as follows:

$$\forall i \in \{1, 2, ..., d\},\ \theta_n^i = \begin{cases} i \cdot \xi & i \in I \\ \theta_{\text{default}} & i \notin I \end{cases} \tag{4}$$

, in which θ_{default} is the default cipher in case of $i \notin I$. We can simply set $\theta_{\text{default}} = -\xi$. During detection, when the next token x_n belongs to the sub-green list of v_i, its detection depends solely on the reconstruction of θ_n^i. The specific form of θ_n^i in Equation 4 enables multiple tokens x_j in the watermark window with hash value $H(x_j) = i$ to serve as equivalent texture keys. Then, following insights from Section 4.1, we can set a small hash space size d to increase the density of equivalent texture keys, thus improving robustness. Meanwhile, the other tokens with $H(x_j) \neq i$ contribute to the construction of other sub-green lists. This design enhances the overall diversity of the green list to $O(2^d)$, allowing SEEK to preserve text quality. The procedure of SEEK is detailed in Algorithm 1.

Algorithm 1 Generation Algorithm

Input: prompt $\{x_1, ..., x_N\}$; integer hash function H with hash space $\{1, 2, ..., d\}$; secret key ξ; watermark strength δ; watermark window size h; large language model P_M.
for $n = N+1, N+2, \ldots$ **do**
  Apply the language model to compute next logit vector $\ell_t \leftarrow P_M(x_{1:n-1})$.
  Calculate the hash signature of the watermark window $I \leftarrow \{H(x_{n-k}) | 1 \leq k \leq h\}$.
  Partition the vocabulary $\mathcal{V}$ into uniform sub-vocabularies $\{v_1, \ldots, v_d\}$ by ξ.
  Derive the cipher θ_n^i for each sub-vocabulary v_i following Eq 4.
  Generate a sub-green list G^i for each sub-vocabulary v_i seeded by the corresponding θ_n^i.
  Union all sub-green lists G^i as the green list G of the full vocabulary.
  Add δ to the logits of tokens in G to modify the distribution by Eq 1.
  Sample the next token from the modified distribution.
end for

5 Experiments

This section presents the main experimental results. Comprehensive details of the experimental setup and hyperparameters are provided in Appendix F, and additional supplementary experiments for robustness under different settings are presented in Appendix B.

Table 2: Robustness of KGW variants and our method against four types of scrubbing attacks across various datasets, evaluated by AUROC (↑), TP@1% (↑), and TP@5% (↑). **Bold** and underlined values indicate the best and second-best performance, respectively.

Dataset		WikiText			C4-Realnewslike			LFQA		
Attacker	**Scheme**	**AUROC**	TP@1%	TP@5%	**AUROC**	TP@1%	TP@5%	**AUROC**	TP@1%	TP@5%
ChatGPT	KGW-Sum^4	80.6	27.2	44.3	75.8	13.9	35.9	81.9	38.7	43.7
	KGW-Min^3	95.9	69.2	86.8	98.3	**83.3**	91.4	96.9	76.1	88.1
	KGW-Min^4	94.9	66.9	84.9	96.6	73.5	86.5	93.3	59.4	75.5
	KGW-Min^6	93.9	60.3	77.2	94.1	52.9	75.0	92.2	50.4	70.2
	KGW-Left^1	93.1	60.1	80.7	94.3	72.8	84.8	92.7	56.2	76.8
	SEEK^6	**96.5**	**69.4**	**90.1**	**98.4**	80.5	**92.6**	**98.2**	**86.4**	**94.4**
Deepseek	KGW-Sum^4	79.8	24.6	41.0	80.3	38.9	44.6	81.7	27.9	48.3
	KGW-Min^3	94.3	32.8	77.4	97.9	74.4	81.0	94.2	55.3	77.0
	KGW-Min^4	95.7	60.6	84.7	96.2	71.7	84.9	92.8	53.4	73.6
	KGW-Min^6	87.3	40.8	60.0	86.3	28.2	52.0	82.5	23.2	45.1
	KGW-Left^1	93.7	52.8	80.3	95.4	70.3	81.4	90.4	48.2	70.8
	SEEK^6	**96.7**	**69.1**	**87.8**	**98.2**	**75.0**	**90.1**	**98.0**	**76.9**	**91.6**
DIPPER-I	KGW-Sum^4	83.1	29.1	49.2	84.2	29.5	51.7	85.7	34.9	54.8
	KGW-Min^3	97.2	76.6	92.1	98.9	92.1	95.8	99.2	94.9	98.1
	KGW-Min^4	94.9	67.4	87.0	97.5	82.6	91.3	98.7	90.8	96.2
	KGW-Min^6	93.7	57.4	77.2	96.4	71.0	83.9	97.4	79.2	89.8
	KGW-Left^1	94.3	64.9	82.6	97.2	80.3	90.1	94.5	88.8	93.4
	SEEK^6	**98.3**	**78.2**	**95.9**	**99.6**	**96.0**	**98.6**	**99.4**	**96.1**	**98.7**
DIPPER-II	KGW-Sum^4	62.6	3.5	14.3	56.7	1.0	10.3	55.5	2.3	10.1
	KGW-Min^3	84.5	20.8	52.4	84.0	28.7	49.8	87.4	34.1	59.2
	KGW-Min^4	80.3	11.4	39.5	82.9	25.4	44.3	81.5	26.0	48.0
	KGW-Min^6	75.7	9.8	26.9	76.8	14.4	34.2	75.7	15.1	34.1
	KGW-Left^1	77.7	13.7	38.1	81.6	24.7	40.3	80.2	27.4	46.8
	SEEK^6	**86.9**	**38.0**	**57.0**	**85.7**	**36.8**	**58.7**	**87.8**	**52.8**	**68.8**

5.1 Experiment Setting

Dataset. We follow previous work [29, 32, 27] to select evaluation datasets. We use C4 [51], WikiText [41], and LFQA datasets [31] to assess watermark robustness against scrubbing. For the spoofing attack, we use C4-Eval [51], Dolly-CW [10], MMW-BookReports, and MMW-FakeNews [49]. Additionally, we evaluate text generation quality on domain-specific datasets, including FreeLaw [6] and PubMedQA [55]. During generation, we randomly sample text segments from these datasets and truncate a fixed-length portion from the end, using the remainder as the prompt for model completion.

Evaluation metrics. We evaluate watermarking methods by assessing both the quality of generated texts and the robustness to both spoofing and scrubbing. To assess the text quality, we employ the perplexity (PPL) using LLaMA2-13B [60] as an oracle, employ the P-SP score [63] between watermarked and non-watermarked texts, and the log-diversity metric [30, 62], following the common setup [72, 29, 30]. For scrubbing robustness, we report AUROC and the true positive rate at false positive rates of f (TP@f). We consider two paraphrase-based scrubbing attacks using prompted ChatGPT-3.5 [7] and DeepSeek-V3 [36], as well as two configurations of DIPPER [31], denoted as DIPPER-I ($O = 60, L = 60$) and DIPPER-II ($O = 20, L = 20$). Regarding spoofing robustness, we use FPR@f [27] to denote the fraction of attacker's texts detected as watermarked by a victim's detector calibrated to a false-positive rate of f on non-adversarial texts. We evaluate against two types of spoofing: a statistics-based attack [27] and a distillation-based attack [18]. For training spoofing methods, we collected 30,000 watermarked responses, each with a maximum length of 800 tokens. We utilize Qwen2-0.5B-Instruct [1] as the initial checkpoint for distillation-based attack.

Baseline and Language Model. Among h-gram watermarking schemes, KGW-Min (i.e. SelfHash [30]) demonstrates the strongest balance between scrubbing and spoofing robustness, making it our primary baseline. We denote watermarking schemes with a window size of h as SCHEME^h. For our experiments, we utilize LLaMA2-7B/13B [60], Mistral-7B [26] and OPT-6.7B [70] language models.

5.2 Overall Performance

Robustness Against Scrubbing Attacks. As shown in Table 2, our method exhibits strong robustness against scrubbing attacks across all evaluation datasets and attackers, achieving TP@1% improvements of +10.8%, +13.4%, and +8.6% on WikiText, C4, and LFQA, respectively, compared

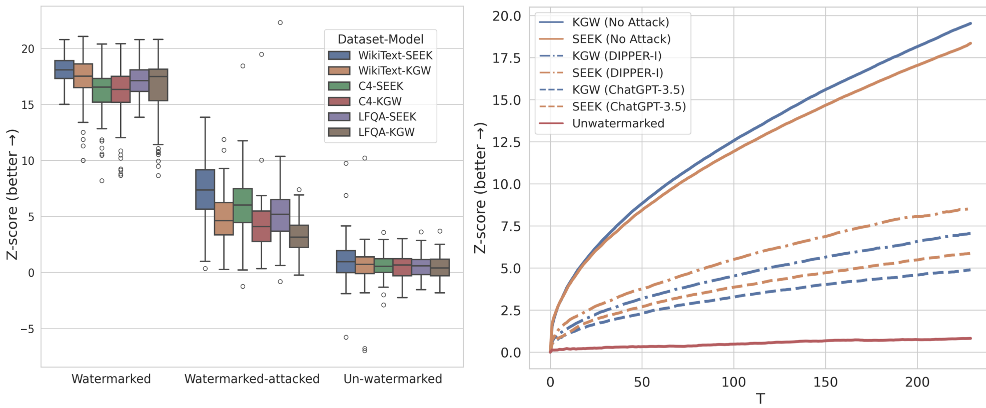

Figure 5: The z-score comparison across watermarking conditions. **(Left)**: Box plots of z-scores for our method and KGW-Min[4] across different datasets. **(Right)**: The advantage z-scores as a function of text length, comparing robustness under different scrubbing attacks.

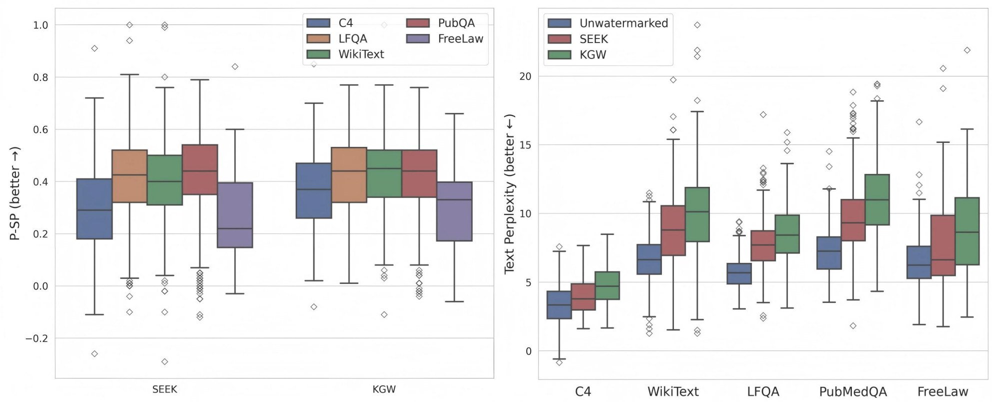

Figure 6: Evaluation of semantic preservation and generation quality of different watermarking methods. **(Left)**: P-SP scores across datasets for two models using our method and KGW-Min[4]. **(Right)**: Text perplexity comparison among unwatermarked, our method, and KGW-Min[4] across various datasets and models. The watermark parameters γ and δ are fixed at $(0.25, 2.0)$.

Table 3: Performance comparison of various SEEK parameter schemes under the DIPPER-I attack on a C4-Eval subset. h and d denote the watermark window size and the cardinality of the hash space, respectively.

Configurations	AUROC ↑	TP@1% ↑	TP@5% ↑
(h=6, d=6)	99.0	95.0	98.1
(h=6, d=8)	98.9	93.8	95.4
(h=8, d=8)	98.9	94.1	95.8
(h=9, d=9)	98.8	94.0	95.6
(h=9, d=6)	99.6	96.8	98.2
(h=12, d=12)	99.1	93.2	95.6

Table 4: Performance of different watermarking schemes on the Log Diversity metric across various datasets, relative to the unwatermarked text. Watermark parameters γ and δ are fixed at $(0.25, 2.0)$.

Schemes	WikiText	LFQA	PubMedQA
UNIGRAM	0.70	0.85	0.80
KGW-Min (d=$\|\mathcal{V}\|$)	0.96	0.95	0.95
KGW-Min (d=4)	0.78	0.90	0.87
KGW-Min (d=16)	0.85	0.91	0.87
SEEK (d=6)	0.92	0.94	0.89
SEEK (d=12)	0.95	0.96	0.95

to KGW-Min[4]. We also investigate the length scaling performance of proposed SEEK. Table 3 reports the performance of various SEEK parameter configurations under the DIPPER-I attack on a C4-Eval subset. We do not recommend using excessively disparate h and d values, which causes the design to degrade to h=0. As shown in Figure 5, the efficacy of SEEK grows with the text length. Moreover, our method consistently exhibits smaller drops in z-score under paraphrasing-based scrubbing attacks compared to KGW, indicating superior scrubbing robustness.

Robustness Against Spoofing Attacks. Table 5 demonstrates that our method consistently achieves lower FPR@f than all baselines across different settings. Notably, KGW-Left[1] exhibits better robustness than KGW-MIN[3], even with a smaller watermark window. We attribute this pheonomenon to the higher z-score of self-seeding, leading to a more informative training corpus for the attacker.

Table 5: Robustness of KGW variants and our method against statistics-based spoofing attacks across three benchmark datasets, evaluated by FPR@1e-3 (↓), FPR@1e-2 (↓), and PPL (↓). **Bold** and underlined values indicate the best and second-best performance, respectively.

Scheme	Dolly-CW			MMW-BookReports			MMW-FakeNews		
	FPR@1e-3	FPR@1e-2	PPL	FPR@1e-3	FPR@1e-2	PPL	FPR@1e-3	FPR@1e-2	PPL
KGW-Left[1]	0.83	0.91	27.7	0.87	0.92	29.4	0.91	0.99	27.4
KGW-Min[3]	0.98	0.99	31.3	0.99	1.00	34.6	1.00	1.00	28.7
KGW-Min[4]	0.85	0.93	25.9	0.91	0.95	26.5	0.95	0.98	24.1
KGW-Min[5]	0.31	0.48	24.8	0.27	0.50	25.4	0.64	0.80	23.4
KGW-Min[6]	<u>0.13</u>	<u>0.19</u>	<u>24.0</u>	<u>0.10</u>	<u>0.24</u>	<u>25.4</u>	<u>0.31</u>	<u>0.67</u>	23.3
KGW-Sum[4]	0.17	0.28	24.7	0.26	0.41	27.7	0.25	0.31	<u>23.2</u>
SEEK	**0.10**	**0.16**	**24.8**	**0.07**	**0.18**	**25.6**	**0.18**	**0.26**	**22.9**

Table 6: Visualization of scrubbing and spoofing attack samples from the Dolly-CW dataset. The p-value indicates how likely the watermark strength could occur by chance under the null hypothesis of the z-test; a smaller value implies stronger evidence of a watermark. Green and red tokens are color-coded, respectively.

Prompt	Imagine you are the last person alive. You're an astronaut and alone and in space aboard the ISS. Write down your feelings and what happened to get there, with a detailed backstory.
(a) Scrubbing Attacked (KGW-Min[4]) p-value: 4.68e-05 z-score: 3.90 (↓ 14.51)	As the last person alive, or at least the last one John knew of, each passing Tuesday aboard the International Space Station blurred into the next. Floating effortlessly in zero gravity had become second nature to him; after two years in orbit, he could perform every operation without so much as a glance. Yet, no amount of routine dulled the quiet wonder he [continues...]
(b) Scrubbing Attacked (SEEK) p-value: 4.57e-10 z-score: 6.14 (↓ 11.25)	As the last person alive, I sit tethered to the viewport of the International Space Station, watching Earth spin slowly below me—beautiful, silent, and lifeless. It's strange how quiet the universe is when you're no longer waiting for a voice. The hum of the life support systems, the soft beeping of monitors, the occasional creak of the station's metal as it contracts [continues...]
(c) Spoofing Attacked (KGW-Min[4]) p-value: 7.25e-59 z-score: 16.13	As the last person alive, I cannot help but reflect on the incredible journey that brought me to this unforeseen fate. As a seasoned astronaut, I have had the privilege of exploring the cosmos, witnessing the beauty of the universe from the ISS, and making groundbreaking scientific discoveries. However, as time passed, the world as we knew it began to collapse, [continues...]
(d) Spoofing Attacked (SEEK) p-value: 0.02 z-score: 1.90	As the last person alive in the vastness of space, I find myself in a state of profound solitude. The International Space Station (ISS) has been my home for what feels like an eternity, a lonely outpost in the cosmos. I am an astronaut, trained to withstand the harsh conditions of space travel and to carry out scientific experiments with precision and accuracy. But [continues...]

Table 7: Comparison of model robustness under distillation-based spoofing attacks [18] between KGW variants and our proposed method across four benchmark datasets, evaluated by z-scores (↓), FPR@1e-2 (↓). **Bold** and underlined values indicate the best and second-best performance, respectively.

Scheme	Dolly-CW		MMW-BookReports		MMW-FakeNews		C4-Eval	
	z-scores	FPR@1e-2	z-scores	FPR@1e-2	z-scores	FPR@1e-2	z-scores	FPR@1e-2
KGW-Min[3]	3.99	0.47	3.93	0.50	6.39	0.85	3.18	0.37
KGW-Min[4]	3.66	0.38	3.48	0.35	4.61	0.66	2.51	0.24
KGW-Min[5]	3.45	0.37	2.99	0.27	3.53	0.34	2.35	0.16
KGW-Min[6]	<u>2.89</u>	<u>0.24</u>	<u>2.68</u>	<u>0.14</u>	<u>2.89</u>	<u>0.17</u>	<u>2.14</u>	<u>0.15</u>
SEEK	**2.17**	**0.11**	**2.42**	**0.09**	**2.55**	**0.16**	**1.97**	**0.12**

To further assess the spoofing robustness of SEEK, we also adopt the distillation-based spoofing [18]. Results in Table 7 consistently show our method's superior robustness across all datasets. Interestingly, we observed a monotonic non-increasing relationship between the watermark window size and resistance to distillation-based spoofing, similar to statistics-based spoofing.

Watermark text generation quality. Figure 6 compares the generation quality across different datasets. Table 4 compares the effectiveness of different watermarking schemes in maintaining the Log Diversity of generated text across multiple datasets. The results demonstrate that our method maintains linguistic fidelity comparable to KGW, with negligible impact on the text quality.

Visualization. Table 6 visualizes the responses of watermarking schemes to scrubbing and spoofing attacks. The scrubbing attack is performed by ChatGPT-3.5, with the z-score in parentheses indicating the decrease caused by the attack, while spoofing robustness is evaluated using statistics-based attacks. Under scrubbing, KGW results in greater loss of watermark evidence and lower z-scores than SEEK, increasing the risk of false negatives. Under spoofing attacks, adversaries can easily mimic KGW watermark patterns, leading to abnormally high z-scores than SEEK.

6 Conclusion

In this work, we discover a mechanism of equivalent texture keys to address the tradeoff problem of existing ones under both scrubbing and spoofing attacks. Based on this, we introduce SEEK, which enables a robust watermark embedding with large token windows without compromising text quality. Experiments across diverse datasets and attack settings demonstrate that our approach outperforms prior methods and achieves a Pareto-optimal balance between scrubbing and spoofing attacks.

7 Acknowledgements

This work was supported by Beijing Natural Science Foundation (L253001) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

[1] Qwen2 technical report. 2024.

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

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

[4] M. J. Atallah, V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed, and S. 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*, pages 185–200. Springer, 2001.

[5] M. J. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg. Natural language watermarking and tamperproofing. In *International workshop on information hiding*, pages 196–212. Springer, 2002.

[6] S. Biderman, K. Bicheno, and L. Gao. Datasheet for the pile. *arXiv preprint arXiv:2201.07311*, 2022.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

[9] M. Conover, M. Hayes, A. Mathur, X. Meng, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, et al. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023.

[10] M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, and R. Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm. `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`, 2023. Accessed: 2023-06-30.

[11] F. Z. Dai and Z. Cai. Towards near-imperceptible steganographic text. *arXiv preprint arXiv:1907.06679*, 2019.

[12] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL `https://doi.org/10.18653/v1/n19-1423`.

[13] J. Fairoze, S. Garg, S. Jha, S. Mahloujifar, M. Mahmoody, and M. Wang. Publicly-detectable watermarking for language models. *arXiv preprint arXiv:2310.18491*, 2023.

[14] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli. ELI5: long form question answering. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3558–3567. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1346. URL `https://doi.org/10.18653/v1/p19-1346`.

[15] S. Gehrmann, H. Strobelt, and A. M. Rush. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*, 2019.

[16] S. S. Ghosal, S. Chakraborty, J. Geiping, F. Huang, D. Manocha, and A. S. Bedi. Towards possibilities & impossibilities of ai-generated text detection: A survey. *arXiv preprint arXiv:2310.15264*, 2023.

[17] E. Giboulot and T. Furon. Watermax: breaking the llm watermark detectability-robustness-quality trade-off. *arXiv preprint arXiv:2403.04808*, 2024.

[18] C. Gu, X. L. Li, P. Liang, and T. Hashimoto. On the learnability of watermarks for language models. *arXiv preprint arXiv:2312.04469*, 2023.

[19] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, and T. Goldstein. Spotting llms with binoculars: Zero-shot detection of machine-generated text, 2024. *URL: https://arxiv. org/abs/2401.12070*.

[20] X. He, Q. Xu, L. Lyu, F. Wu, and C. Wang. Protecting intellectual property of language generation apis with lexical watermark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10758–10766, 2022.

[21] X. He, Q. Xu, Y. Zeng, L. Lyu, F. Wu, J. Li, and R. Jia. Cater: Intellectual property protection on text generation apis via conditional watermarks. *Advances in Neural Information Processing Systems*, 35:5431–5445, 2022.

[22] A. B. Hou, J. Zhang, T. He, Y. Wang, Y.-S. Chuang, H. Wang, L. Shen, B. Van Durme, D. Khashabi, and Y. Tsvetkov. Semstamp: A semantic watermark with paraphrastic robustness for text generation. *arXiv preprint arXiv:2310.03991*, 2023.

[23] A. B. Hou, J. Zhang, Y. Wang, D. Khashabi, and T. He. k-semstamp: A clustering-based semantic watermark for detection of machine-generated text. *arXiv preprint arXiv:2402.11399*, 2024.

[24] D. Hovy. The enemy in your own camp: How well can we detect statistically-generated fake reviews–an adversarial study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 351–356, 2016.

[25] B. Huang, X. Pu, and X. Wan. b^4: A black-box scrubbing attack on llm watermarks. *arXiv preprint arXiv:2411.01222*, 2024.

[26] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

[27] N. Jovanović, R. Staab, and M. Vechev. Watermark stealing in large language models. *arXiv preprint arXiv:2402.19361*, 2024.

[28] A. Kerckhoffs. La cryptographie militaire. *J. Sci. Militaires*, 9(4):5–38, 1883.

[29] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR, 2023.

[30] J. Kirchenbauer, J. Geiping, Y. Wen, M. Shu, and K. Saifullah. On the reliability of watermarks for large language models. *arXiv preprint arXiv:2306.04634*, 2023.

[31] K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36:27469–27500, 2023.

[32] R. Kuditipudi, J. Thickstun, T. Hashimoto, and P. Liang. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

[33] T. Lee, S. Hong, J. Ahn, I. Hong, H. Lee, S. Yun, J. Shin, and G. Kim. Who wrote this code? watermarking for code generation. *arXiv preprint arXiv:2305.15060*, 2023.

[34] A. Liu, L. Pan, X. Hu, S. Li, L. Wen, I. King, and P. S. Yu. An unforgeable publicly verifiable watermark for large language models. *arXiv preprint arXiv:2307.16230*, 2023.

[35] A. Liu, L. Pan, X. Hu, S. Meng, and L. Wen. A semantic invariant robust watermark for large language models. *ArXiv*, abs/2310.06356, 2023. URL `https://api.semanticscholar.org/CorpusID:263830310`.

[36] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[37] A. Liu, L. Pan, Y. Lu, J. Li, X. Hu, X. Zhang, L. Wen, I. King, H. Xiong, and P. Yu. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36, 2024.

[38] Y. Lu, A. Liu, D. Yu, J. Li, and I. King. An entropy-based text watermarking detection method. *arXiv preprint arXiv:2403.13485*, 2024.

[39] Y. Lu, A. Liu, D. Yu, J. Li, and I. King. An entropy-based text watermarking detection method. *arXiv preprint arXiv:2403.13485*, 2024.

[40] M. Mao, D. Wei, Z. Chen, X. Fang, and M. Chau. A watermark for low-entropy and unbiased generation in large language models. *arXiv preprint arXiv:2405.14604*, 2024.

[41] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models, 2016.

[42] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR, 2023.

[43] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR, 2023.

[44] OpenAI. ChatGPT, 2022. URL `https://chat.openai.com`.

[45] OpenAI. GPT-4 Technical Report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/arXiv.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

[46] Q. Pang, S. Hu, W. Zheng, and V. Smith. Attacking llm watermarks by exploiting their strengths. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024.

[47] Q. Pang, S. Hu, W. Zheng, and V. Smith. No free lunch in llm watermarking: Trade-offs in watermarking design choices. *arXiv preprint arXiv:2402.16187*, 2024.

[48] Q. Pang, S. Hu, W. Zheng, and V. Smith. Attacking llm watermarks by exploiting their strengths. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024.

[49] J. Piet, C. Sitawarin, V. Fang, N. Mu, and D. Wagner. Mark my words: Analyzing and evaluating language model watermarks. *arXiv preprint arXiv:2312.00273*, 2023.

[50] S. Pudasaini, L. Miralles-Pechuán, D. Lillis, and M. Llorens Salvador. Survey on ai-generated plagiarism detection: The impact of large language models on academic integrity. *Journal of Academic Ethics*, pages 1–34, 2024.

[51] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[52] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[53] J. Ren, H. Xu, Y. Liu, Y. Cui, S. Wang, D. Yin, and J. Tang. A robust semantics-based watermark for large language model against paraphrasing. *arXiv preprint arXiv:2311.08721*, 2023.

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

[55] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective Classification in Network Data. *AI Magazine*, 29(3):93, Sept. 2008. doi: 10/ggn9hw. URL https://www.aaai.org/ojs/index.php/aimagazine/article/view/2157.

[56] K. Stefan, P. Fabien AP, et al. Information hiding techniques for steganography and digital watermarking, 2000.

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

[58] U. Topkara, M. Topkara, and M. J. Atallah. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of the 8th workshop on Multimedia and security*, pages 164–174, 2006.

[59] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

[60] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton-Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

[61] L. Wang, W. Yang, D. Chen, H. Zhou, Y. Lin, F. Meng, J. Zhou, and X. Sun. Towards codable watermarking for injecting multi-bits information to llms. *arXiv preprint arXiv:2307.15992*, 2023.

[62] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.

[63] J. Wieting, K. Gimpel, G. Neubig, and T. Berg-Kirkpatrick. Paraphrastic representations at scale. *arXiv preprint arXiv:2104.15114*, 2021.

[64] C. Woodcock. AI Is Tearing Wikipedia Apart, May 2023. URL `https://www.vice.com/en/article/v7bdba/ai-is-tearing-wikipedia-apart`. Accessed: 2025-03-30.

[65] Q. Wu and V. Chandrasekaran. Bypassing llm watermarks with color-aware substitutions. *arXiv preprint arXiv:2403.14719*, 2024.

[66] Y. Wu, Z. Hu, J. Guo, H. Zhang, and H. Huang. A resilient and accessible distribution-preserving watermark for large language models. *arXiv preprint arXiv:2310.07710*, 2023.

[67] K. Yoo, W. Ahn, and N. Kwak. Advancing beyond identification: Multi-bit watermark for large language models. *arXiv preprint arXiv:2308.00221*, 2023.

[68] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019.

[69] H. Zhang, B. L. Edelman, D. Francati, D. Venturi, G. Ateniese, and B. Barak. Watermarks in the sand: Impossibility of strong watermarking for generative models. *arXiv preprint arXiv:2311.04378*, 2023.

[70] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[71] Z. Zhang, X. Zhang, Y. Zhang, L. Y. Zhang, C. Chen, S. Hu, A. Gill, and S. Pan. Large language model watermark stealing with mixed integer programming. *arXiv preprint arXiv:2405.19677*, 2024.

[72] X. Zhao, P. Ananth, L. Li, and Y.-X. Wang. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*, 2023.

[73] Z. M. Ziegler, Y. Deng, and A. M. Rush. Neural linguistic steganography. *arXiv preprint arXiv:1909.01496*, 2019.

Warning: This appendix contains unfiltered content generated by LLMs that may be offensive to some readers.

A Full Related Work

Machine-human generation detection. The task of distinguishing between machine-generated and human-written text can be addressed through two primary approaches: post-hoc detection and proactive detection. Post-hoc detection operates independently of the text generation process and generally leverages a priori knowledge about the statistics of texts generated by a given class of LLMs. For instance, GLTR [15] estimates the likelihood of individual tokens and applies thresholding techniques to detect AI-generated text. DetectGPT [42] posits that such text typically resides in regions of negative curvature within the log-likelihood landscape. As language models grow more advanced, their outputs closely mimic human writing, making detection increasingly difficult. Existing statistical detectors often require white-box access to know the language model statistics, such as perplexity or curvature [30]. Despite their flexibility, these methods exhibit limited performance, with error rates empirically validated on limited datasets and rarely falling below 10^{-3} [19, 17]. Proactive detection must be actively deployed by model owners, yet it consistently outperforms passive detection in both effectiveness and reliability. A prominent example is watermarking, which enables the embedding of imperceptible signals into text while preserving its overall quality.

LLM Watermarking technology. Watermarking embeds identifying information within data, but its application to natural language faces challenges due to its discrete and symbolic nature [56]. Early methods employed rule-based techniques such as syntactic restructuring [4], synonym substitution [58], and paraphrasing [5]. More recent neural network watermarking approaches [21, 73, 11, 3, 20] are introduced to develop a black-box watermark that adopts end-to-end learning frameworks during the generation and detection process. However, their limited theoretical guarantees and lack of interpretability hinder broader adoption in practice. Early foundational pioneering work in LLM watermarking includes KGW [29, 30] and AAR [2] place watermarks on a robust mathematical foundation with a low and provable false positive rate. KGW introduces the previous context as the watermark window to modify the next token's distribution, which significantly influenced numerous follow-up watermark works [32, 40, 33, 72, 34, 39]. Building on this, watermark detection is later formulated as a hypothesis testing problem, where the alternative hypothesis assumes that the text is drawn from the modified distribution. Subsequent work has extended this line of research by incorporating semantic information [22, 53], embedding multi-bit information [61, 67], and exploring watermarking in low-entropy settings such as code generation [33, 38]. The UNIGRAM [32] applies fixed green lists to remove contextual dependency, emphasizing robustness to scrubbing over the comparatively minor threat of spoofing. This line of research has further evolved to incorporate semantic information [22, 53], embedding multi-bit information [61, 67], and exploring watermarking in low-entropy settings such as code generation [33, 38]. Our work focuses on analyzing the security properties and robustness of KGW-based watermarking frameworks

B Additoinal Experiment Results

B.1 Different Foundation Models

Table 8 compares the detection performance of various watermarking schemes under GPT-3.5-based scrubbing attacks applied to the C4 dataset, across three different language models: LLaMA-13B, Mistral-7B, and OPT-6.7B. The results demonstrate that our proposed methods consistently outperform the KGW-Min baselines in terms of AUROC and true positive rates at both low (1%) and moderate (5%) false positive rates. Importantly, our method maintains high detection accuracy across all model backbones, with AUROC values exceeding 97% in most cases and TP@5% reaching over 90%, even under aggressive scrubbing. This robustness indicates strong generalization to different model architectures and capacities. For instance, while KGW-Min variants show significant degradation in TP rates on smaller models like OPT-6.7B (e.g., TP@5% dropping below 70% for KGW-Min^5 and ^{6}), our approach sustains high recall (e.g., 96.4% for SEEK on OPT-6.7B), illustrating resilience to both content-level and model-level variability. These results suggest that our watermarking framework is not only effective under direct attacks but also exhibits strong cross-model generalization.

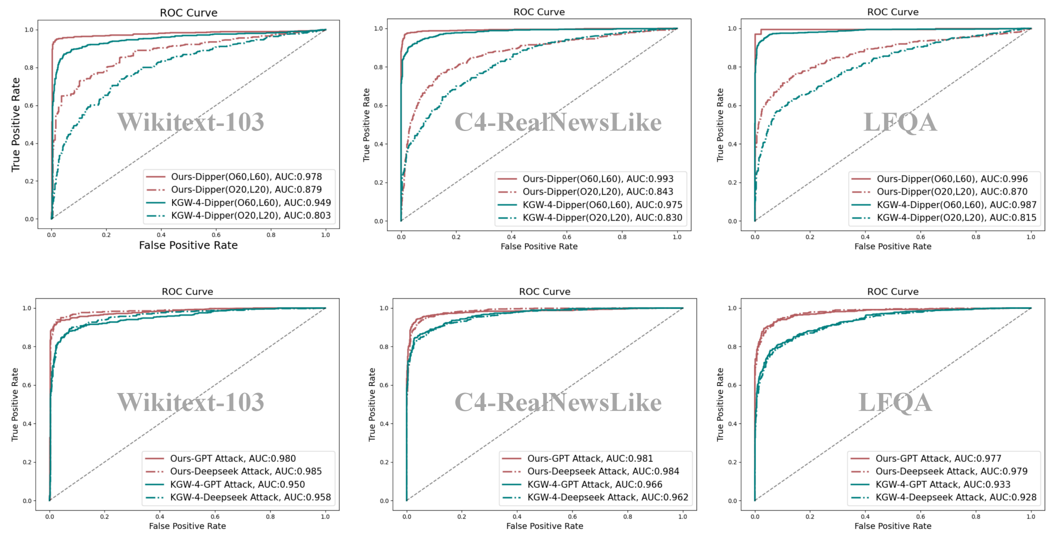

Figure 7: ROC curves comparing our proposed watermarking method with the KGW baseline under various attack scenarios and datasets. Each plot illustrates the detection performance in terms of the true positive rate versus false positive rate. The top row shows results under the DIPPER paraphrasing attack with two parameter settings, while the bottom row presents results under GPT and Deepseeker attacks. Evaluation is conducted across three benchmark datasets: Wikitext (left), C4-RealNewsLike (center), and LFQA (right). Our method consistently achieves higher AUROC scores across all conditions, demonstrating superior scrubbing robustness compared to KGW.

Table 8: Comparative analysis of watermark robustness under scrubbing attacks across multiple language models. All watermarking schemes are subjected to ChatGPT-3.5-based scrubbing on the C4-RealNewsLike dataset. Our proposed methods are compared against a series of KGW-Min baselines, demonstrating consistently stronger resilience to scrubbing attack.

Scheme	**LLAMA-13B**			**Mistral-7B**			**OPT-6.7B**		
	AUROC	TP@1%	TP@5%	**AUROC**	TP@1%	TP@5%	**AUROC**	TP@1%	TP@5%
KGW-Min^3	97.2	74.4	89.5	96.2	71.9	85.3	96.3	73.4	85.9
KGW-Min^4	95.9	65.2	82.6	94.2	58.1	76.3	93.2	61.8	74.7
KGW-Min^5	95.1	58.9	79.9	90.8	53.5	69.1	90.8	54.0	68.2
KGW-Min^6	93.7	62.4	77.8	90.1	47.1	61.6	89.5	47.6	63.5
SEEK	97.9	79.4	90.9	98.7	90.2	94.9	99.0	89.4	96.4

B.2 Performance under Beam Search

Table 9 presents the detection performance of various watermarking schemes under ChatGPT-3.5-based scrubbing attacks, evaluated on the C4 dataset using two beam search configurations: beam width 2 and beam width 4. Although increasing the beam search width generally improves the fluency and quality of the generated text, it also makes watermark embedding more challenging. Across both settings, our proposed methods outperform the KGW-Min baselines, particularly in terms of true positive rate at 5% false positive rate (TP@5%). Notably, SEEK$_L$ achieves the highest detection scores in both configurations. These results suggest that our watermarking approach is robust to variations in decoding strategies and can generalize well under different text generation conditions, making it suitable for real-world deployment where generation parameters may vary dynamically.

B.3 Copy-paste Attacks

We simulate three Copy-Paste (CP) attack configurations, denoted as CP-M-P%. For example, CP-1-25% indicates that 25% of the document is replaced with watermarked content inserted at a single contiguous location. Table 10 presents a comparative evaluation of detection performance under different Copy-Paste attack configurations across three sliding window-based detectors (Win20, Win40, and WinMax). When employing the WinMax configuration, which selects the best performance across all tested window sizes, both watermarking schemes demonstrate strong resilience

Table 9: Evaluation of watermark robustness under GPT-3.5-based scrubbing attacks on the C4 dataset across different sampling strategies. The comparison includes our proposed methods and KGW-Min baselines under beam search decoding with beam widths of 2 and 4. Results show that across various paraphrasers and datasets, our method outperforms KGW-based watermarking.

Scheme	Beam search=2			Beam search=4		
	AUROC	TP@1%	TP@5%	AUROC	TP@1%	TP@5%
KGW-Min^3	93.1	25.9	73.4	92.4	41.5	65.1
KGW-Min^4	91.6	42.4	63.1	90.5	31.2	70.8
KGW-Min^5	90.6	31.9	60.8	90.8	20.1	61.8
KGW-Min^6	89.8	33.7	57.9	90.3	21.0	53.8
SEEK	93.6	41.6	81.3	92.7	38.8	66.9

Table 10: Comparative analysis of detection performance under different Copy-Paste (CP) attack settings. Each CP format is denoted as CP-M-$P\%$, where M indicates the number of disjoint positions in the document containing watermarked content, and $P\%$ represents the total proportion of watermarked text. **Win20** and **Win40** denote sliding window detectors with window sizes of 20 and 40 tokens, respectively, while **WinMax** represents the best detection performance across all tested window sizes. We compare our method with the KGW-Min baseline across multiple detectors and evaluate performance using AUROC, TP@1%, and TP@5%.

Metrics		CP-1-25%			CP-1-10%			CP-3-25%		
Detector	**Scheme**	**AUROC**	TP@1%	TP@5%	**AUROC**	TP@1%	TP@5%	**AUROC**	TP@1%	TP@5%
Win20	KGW-Min^4	99.7	99.7	99.8	97.1	99.0	99.0	99.0	98.1	99.6
	SEEK	99.8	99.6	99.7	98.9	96.1	99.4	99.1	98.3	99.8
Win40	KGW-Min^4	99.9	99.7	100	96.7	84.6	98.6	97.4	86.0	98.8
	SEEK	99.8	99.8	100	97.1	88.4	97.1	96.9	87.4	98.1
WinMax	KGW-Min^4	100	100	100	99.6	99.7	99.0	99.5	97.8	99.4
	SEEK	100	100	100	99.7	98.7	99.8	99.7	98.7	99.8

against CP attacks, achieving near-perfect detection in most scenarios. This highlights the scalability and adaptability of our approach across different detector granularities.

C Algorithm

C.1 Self-seeding Strategy

The self-seeding strategy is introduced in [30, 29] to incorporate the next token x_t as part of the watermark window and generate the green list, which effectively increases the context width by 1. Initially, the language model computes the logits for the next token based on the previous tokens, sorting them in descending order to identify the most likely candidate. The algorithm pre-samples the most probable next token as part of for the construction of the green and red lists. If the final selected next token is not present in the vocabulary, the current sampling attempt is discarded, and the next most probable token is chosen. In practice, the number of sampling attempts k is typically constrained by a maximum threshold (e.g., 40). The overall algorithm is described in Algorithm 2.

Algorithm 2 Self-seeding Strategy Algorithm

Input: watermark strength δ; input prompt $x_{1:n-1}$, a large language model P_M.
Apply the language model to compute next logit vector $\ell_t \leftarrow P_M(x_{1:n-1})$.
Sort the ℓ_t in descending order. Set $k = 0$, assigning the index of the current most likely token.
for $k = 0, 1, \ldots$ **do**
    Generate the green list G and red list R.
    if $\ell_t[k]$ in G **then**
        choose $\ell_t[k]$ as next token and break .
    else if $\ell_t[k]$ in R, and $\ell_t[k] + \delta < \ell_t[0]$ **then**
        choose $\ell_t[0]$ as next token and break .
    else
        Set $k \leftarrow k + 1$, continue.
    end if
end for

C.2 KGW Watermark Variants

When generating the next token x_n, the watermark window is instantiated as a predefined subsequence $x_{n-h:n-1}$, where h is the size of the watermark window. Then tokens in the watermark window are processed using aggregation functions $f : \mathbb{N}^h \rightarrow \mathbb{N}$ (or $f : \mathbb{N}^{h+1} \rightarrow \mathbb{N}$ when employing self-seeding) to seed a pseudorandom function P. The resulting hash function is denoted as H. There are three notable variants of f:

- **KGW-Sum**: This function aggregates the sum of the tokens in the context x, defined as $f_{\text{Sum}}(x) = H\left(\xi \cdot \sum_{i=1}^{h} x_{n-i}\right)$. This approach exhibits strong robustness against spoofing attacks. However, it is sensitive to any permutation in the context x, where even small changes, such as token swaps or removals, alter the resulting hash, thereby breaking the watermark integrity.
- **KGW-Skip**: This variant processes only the leftmost token in the window, formulated as $f_{\text{Skip}}(x) = H(\xi \cdot x_{n-h})$, where x_h is the leftmost token. This function is robust to changes in tokens other than the leftmost one but is vulnerable to insertion and deletion attacks within the context.
- **KGW-Min**: This variant computes the minimum of the Hash values generated for each token in the context. Defined as$f_{\text{Min}}(x) = \xi \cdot \min_{i\in 1,\ldots,h} H(x_{n-i})$. It is robust to permutations and is partially resistant to insertion and deletion. Given that each token produces a pseudo-random value for vocabulary partition, the likelihood of failure is minimized. For instance, in a context with $h = 4$ and two missing tokens, the scheme still has a 50% chance of generating the same hash.

C.3 Watermark Detection

We now consider the detection problem, which involves determining whether a given text is watermarked or not. Specifically, given a sequence $\tilde{x}_{1:|T|} = \{x_1, \ldots, x_T\}$, which may potentially be watermarked, the detection process calculates a z-statistic under the null hypothesis, as follows:

H_0 : The text sequence $\tilde{x}_{1:T}$ is generated with no knowledge of the green-red list rule.

Due to the random selection process of the red list, a natural writer is expected to violate the red list rule with approximately γ of their tokens. In contrast, the watermarked model produces no violations. The probability that a natural source generates T tokens without violating the red list rule is γ^T, which becomes vanishingly small even for short text fragments with only a dozen words [29]. If the null hypothesis holds, the number of green list tokens, denoted $|s|_G$, has an expected value of $T\gamma$ and variance $T\gamma(1-\gamma)$. Then the z-statistic score is computed using the formula:

$$z = \frac{(|s|_G - \gamma T)}{\sqrt{T\gamma(1-\gamma)}}. \tag{5}$$

Under the null hypothesis, where the sequence is assumed not to be watermarked, the detector evaluates whether x is generated by a specific language model by comparing the computed z statistic score to a threshold $z_{threshold}$. If $z > z_{threshold}$ holds, we decide that the watermark is embedded in x and thus generated by the LLM.

The work in [30] introduces a novel threat scenario: an attacker can conceal a watermarked paragraph embedded inside a much larger non-watermarked document, effectively circumventing traditional z-test detection methods. The original z-test, as described in Equation 5, may not be optimal. In such cases, a sliding window detection method known as WinMax is typically employed to identify the watermarked intervals within the given text. Given a sequence of tokens, we first score the sequence on the per-token basis to find the binary vector of hits $s \in \{0,1\}^T$ to each green list, which we can convert to a partial sum representation $p_k = \sum_{i=1}^{k} s_i$. WinMax then searches for the continuous span of tokens that generates the highest z-score. More formally, it computes the following expression:

$$z_{\text{win-max}} = \max_{i,j,i<j} \frac{(p_j - p_i) - \gamma(j-i)}{\sqrt{\gamma(1-\gamma)(j-i)}} \tag{6}$$

However, relying solely on WinMax makes the system highly susceptible to spoofing attacks. An attacker can easily generate a high z-score watermarked text by inserting a watermarked segment into arbitrary malicious content. To mitigate this vulnerability, we recommend that watermark detectors simultaneously assess both the z and the $z_{\text{win-max}}$ values, incorporating multiple layers of defense and strategies tailored to various attack types.

C.4 KGW with Equivalent Texture Keys

Algorithm 3 KGW Generation Algorithm

Input: prompt $\{x_1, ..., x_N\}$; integer hash function H with hash space $\{1, 2, ..., d\}$; secret key ξ; watermark strength δ; watermark window size h; large language model P_M.
for $n = N+1, N+2, \dots$ **do**
  Apply the language model to compute a logit vector $\ell_n \leftarrow P_M(x_{1:n-1})$.
  Definde cipher θ_n satisfying:

$$\theta_n \leftarrow \min\{H(x_{n-h}), H(x_{n-h+1}) \dots, H(x_{n-1})\} \cdot \xi \tag{7}$$

  Partition the vocabulary $\mathcal{V}$ into G and R seeded by θ_n.
  Add δ to the logits of tokens in G, the resulting logits $\hat{\ell}_n[t]$ satisfy:

$$\hat{\ell}_n[t] = \ell_n[t] + \delta \cdot \mathbf{1}_G(t) \tag{8}$$

  Use the watermarked distribution $\hat{\ell}_n[v]$ to sample the next token x_n.
end for

Proposition C.1. *Given a max hash range value d and watermark window size h, the probability of a hash collision occurring can be approximated by $p(h, d) \geq 1 - e^{\frac{-h(h-1)}{2d}}$.*

Proof: Let A denote the event that no collisions occur among h independent samples drawn uniformly at random from a hash space of size d. Then,

$$p(h, d) = \Pr[\overline{A}] = 1 - \Pr[A] \tag{9}$$

The probability of all samples being distinct is given by:

$$\Pr[A] = \prod_{i=0}^{h-1} \left(1 - \frac{i}{d}\right) \tag{10}$$

For $h < d$, we apply the first-order Taylor approximation $1 - x \leq e^{-x}$, obtaining:

$$\Pr[A] \leq \exp\left(-\frac{h(h-1)}{2d}\right) \tag{11}$$

and thus,

$$p(h, d) \geq 1 - e^{\frac{-h(h-1)}{2d}} \tag{12}$$

Proposition C.2. *Given a max hash range value d and watermark window size h, the expected number of token X that must be erased to eliminate the watermark is expressed as $\sum_{m=1}^{d} \frac{h}{d} \left(\frac{d-m+1}{d}\right)^{h-1}$.*

Proof: Let the hash space be denoted by $\{1, 2, \dots, d\}$, and assume that h independent hash values are generated uniformly at random from this space. The random variables $Y_1, Y_2, \dots, Y_h$ represent the values of these h tokens. We aim to compute the expected number X of tokens that must be erased in order to completely eliminate the watermark. Specifically, X represents the number of occurrences of the smallest value among the h hash values, i.e., the number of tokens equal to the minimum value m. Let the minimum value of the set $\{Y_1, Y_2, \dots, Y_h\}$ be denoted as m, and let X represent the number of times the minimum value occurs among the h values. To compute the expectation $\mathbb{E}[X]$, observe that:

$$\mathbb{E}[X] = \sum_{i=1}^{h} \mathbb{E}[\mathbf{1}_{Y_i = m}] \tag{13}$$

where $\mathbf{1}_{Y_i=m}$ is an indicator function that takes the value 1 if $Y_i = m$, and 0 otherwise. Since the tokens are generated independently and identically distributed (i.i.d.) with uniform probability, it follows that each token Y_i has the same probability of being equal to the minimum value m. Thus, we compute the probability that a single token Y_i equals m:

$$\Pr(Y_i = m) = \frac{1}{d}\left(\frac{d-m+1}{d}\right)^{h-1} \tag{14}$$

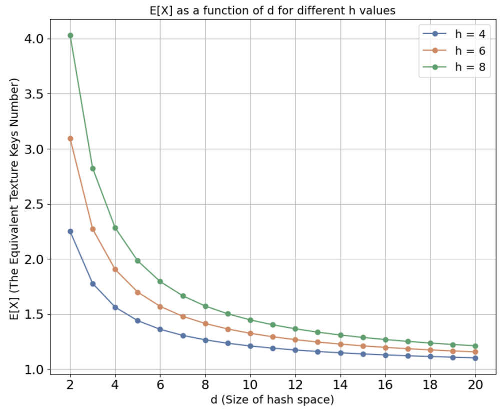

Figure 8: The expected number of equivalent texture keys ($\mathbb{E}[X]$) in watermarking schemes with varying watermark window sizes h and hash space sizes d.

Therefore, the expected number of tokens equal to m is:

$$\mathbb{E}[X] = \sum_{m=1}^{d} \frac{h}{d} \left(\frac{d-m+1}{d} \right)^{h-1} \tag{15}$$

Figure 8 demonstrates the theoretical behavior of watermarking schemes where the number of equivalent texture keys is evaluated for different values of the watermark window size h (4, 6, and 8) and hash space sizes d. Increasing h results in a higher expected number of texture keys, with a more pronounced reduction in the number of equivalent texture keys as d increases. The trend reflects the distribution of the minimum values across different hash space sizes and watermark window sizes, emphasizing the relationship between hash space size and the number of texture key tokens used to protect the watermark.

D Roubutness Guarantees

We discuss the effect of modifying a single token on the watermark removal. For a sequence of tokens $y = \{y_1, \dots, y_n\}$, where all tokens are watermarked (Note that in practice, the text sequence often balances between watermarking and text quality, meaning that the watermark proportion is typically correlated with a parameter δ). Suppose that at position i, a token y_i is inserted, deleted, or modified, resulting in a new token u_i at that position. Let the number of watermarks removed after this modification be represented by $X = \{0, 1, 2, \dots, h\}$. Given the large vocabulary size $|\mathcal{V}|$, we can assume that, under normal circumstances, $u_i \neq y_i$.

Proposition D.1. *For a KWG-MIN with a window size of h, suppose that the hash value of u_i ranks in the bottom $\Phi = \phi$ percentile among the hash values of all tokens in the vocabulary $\mathcal{V}$. Then the probability of having exactly X watermarks removed is given by:*

$$P(X = x|\phi) = \sum_{i=x}^{h} \sum_{k=x}^{i} (1-\phi)^{i-k} \phi^{k+h-1} \binom{k}{x} (1-\gamma)^x \gamma^{k-x} \tag{16}$$

Proof: For a watermark sequence, in order to modify y_o to u_o such that it influences the watermarks in the surrounding context, there must exist a subinterval of length h within the window $\{y_{o-h}, \dots, y_o, \dots, y_{o+h}\}$, where the hash value of each token within the subinterval is smaller than the hash value of u_o.

Let $I = i$ denote the number of consecutive tokens to the left of u_o, starting from y_{o-1}, whose hash values are smaller than $H(u_o)$. We define the first watermark window as $\{y_{o-i}, \dots, u_o, \dots, y_{o+h-i-1}\}$, where i is equal to the maximum sliding distance of the window. If $X > 0$, then there exists at least one interval of length h, where every token within the interval has a hash value smaller than $H(u_o)$. Thus, we first have the basic probability ϕ^{h-1}, which represents the probability of finding a watermark window where all tokens' hash values are smaller than $H(u_o)$.

For the case where $X = x$, we need to ensure that this interval can move to the right by at least x positions. Clearly, $i \geq x$ is required to influence the subsequent x tokens. We now consider the suffix, which is the distribution of $\{y_o, \ldots, y_{o+h-i-1}\}$. Suppose there are K tokens in the suffix whose hash values are smaller than $H(u_o)$, then these $K = k$ tokens will be affected by the modification of y_o to u_o, with probability $P(K = k|I = i) = \phi^{h-1} \cdot (1-\phi)^{i-k}\phi^k$.

For each affected token, there is a probability of $1 - \gamma$ that it belongs to the red list R (i.e., the watermark is removed), and a probability of γ that it remains in the "green" set (i.e., the watermark is not removed). Given $K = k$, the value of X follows a binomial distribution $\text{Binomial}(k, 1-\gamma)$. The probability of removing x watermarks from the k affected tokens is given by:

$$P(X = x|K = k) = \binom{k}{x}(1-\gamma)^x\gamma^{k-x} \tag{17}$$

Given $I = i$, for all possible values of $k \in \{x, x+1, \ldots, i\}$, we define $P(X = x|I = i)$ as the probability of removing x watermarks, which, by the law of total probability, is:

$$P(X = x|I = i) = \sum_{k=x}^{i} P(X = x|I = i, K = k) = \sum_{k=x}^{i} P(X = x|K = k)P(K = k|I = i) \tag{18}$$

Substituting the expressions for $P(X = x|K = k)$ and $P(K = k|I = i)$:

$$P(X = x|I = i) = \phi^{h-1} \times \sum_{k=x}^{i}(1-\phi)^{i-k}\phi^k\binom{k}{x}(1-\gamma)^x\gamma^{k-x} \tag{19}$$

Similarly, by summing over all possible values of $i \in \{x, x+1, \ldots, h\}$, we obtain the final expression for $P(X = x)$ using the law of total probability:

$$\begin{aligned}
P(X = x) &= \sum_{i=x}^{h} P(X = x|I = i) && (20)\\
&= \sum_{i=x}^{h} \phi^{h-1} \times \sum_{k=x}^{i}(1-\phi)^{i-k}\phi^k\binom{k}{x}(1-\gamma)^x\gamma^{k-x} && (21)\\
&= \sum_{i=x}^{h}\sum_{k=x}^{i}(1-\phi)^{i-k}\phi^{k+h-1}\binom{k}{x}(1-\gamma)^x\gamma^{k-x} && (22)
\end{aligned}$$

We further generalize the hypothesis. If we treat $\Phi = \phi$ as a variable, we need to consider all possible values for the inserted token. For simplicity, let $p = \frac{v}{|\mathcal{V}|}$, where v is the index of the token in the vocabulary $\mathcal{V}$:

$$\hat{P} = \sum_{\phi} P(X = x|\Phi = \phi) = \sum_{v=1}^{|\mathcal{V}|}\sum_{i=x}^{h}\sum_{k=x}^{i}(1-p)^{i-k}p^{k+h-1}\binom{k}{x}(1-\gamma)^x\gamma^{k-x} \tag{23}$$

Proposition D.2. *For our watermark scheme with a window size of* h *and hash space* d*, the probability of having exactly* X *watermarks removed is given by:*

$$P(X = x) = \sum_{i=x}^{h}\sum_{k=x}^{i}(\frac{1}{d})^{i-k}(1-\frac{1}{d})^{h+k-1}\binom{k}{x}(\frac{1-\gamma}{d})^x(1-\frac{1-\gamma}{d})^{k-x} \tag{24}$$

Proof: We define the following events:

- **Event A**: The watermark window does not include any texture key to protect the watermark text.
- **Event B**: The watermark window contains only one texture key to protect the watermark text.
- **Event C**: More than two tokens from the watermark window are involved in the hash space that protects the watermark text. In this case, modifying a single token will not remove the watermark.

Events A and C only contribute to $P(X=0)$. Therefore, when considering perturbations for $X>0$, we only need to focus on Event B. We define $I=i$ as the number of consecutive tokens, starting from y_{o-1}, that satisfy the condition $H(y) \neq H(u_o)$, where u_o represents the position of the watermark. The sequence $\{y_{o-i}, \ldots, u_o, \ldots, y_{o+h-i-1}\}$ represents the first watermark window, where i is the maximum sliding distance for this window. The basic probability of a watermark window where no token's hash value equals $H(u_o)$ is given by $\left(1-\frac{1}{d}\right)^{h-1}$, where m is the size of the hash space. For $X=x$, there must exist a segment within the watermark window that can slide to the right by at least x positions. This implies that $i \geq x$ for the watermark to affect subsequent tokens. Next, we consider the suffix $\{y_o, \ldots, y_{o+h-i-1}\}$ and the probability distribution $P(K=k|I=i)$, where K denotes the number of tokens in the suffix that do not match $H(u_o)$. This probability follows:

$$P(K=k|I=i) = \left(1-\frac{1}{d}\right)^{h-1} \cdot \left(\frac{1}{d}\right)^{i-k} \cdot \left(1-\frac{1}{d}\right)^{k} \tag{25}$$

Let Z represent the number of tokens that are affected by u_o. Given $Z=z$, the probability $P(Z=z|K=k)$ follows a binomial distribution:

$$P(Z=z|K=k) = \binom{k}{z}\left(\frac{1}{d}\right)^{z}\left(1-\frac{1}{d}\right)^{k-z} \tag{26}$$

For each affected token, there is a probability of $1-\gamma$ of being in the red set (i.e., the watermark is erased) and γ for being in the green set (i.e., the watermark remains). Given $Z=z$, the value X follows a binomial distribution $\text{Binomial}(z, 1-\gamma)$, and the probability of erasing x watermarks is:

$$P(X=x|Z=z) = \binom{z}{x}(1-\gamma)^{x}\gamma^{z-x} \tag{27}$$

We now compute the overall probability:

$$P(X=x|K=k) = \sum_{z=x}^{k} P(X=x|Z=z)P(Z=z|K=k) \tag{28}$$

$$= \sum_{z=x}^{k} \binom{k}{z}\left(\frac{1}{d}\right)^{z}\left(1-\frac{1}{d}\right)^{k-z}\binom{z}{x}(1-\gamma)^{x}\gamma^{z-x} \tag{29}$$

Using the law of total probability, we can calculate $P(X=x|I=i)$ by summing over k:

$$P(X=x|I=i) = \sum_{k=x}^{i} P(X=x|K=k)P(K=k|I=i) \tag{30}$$

Finally, by iterating over all possible values of i, we obtain the total probability for $X=x$:

$$P(X=x) = \sum_{i=x}^{h} P(X=x|I=i) \tag{31}$$

$$= \sum_{i=x}^{h}\sum_{k=x}^{i}(\frac{1}{d})^{i-k}(1-\frac{1}{d})^{h+k-1}\sum_{z=x}^{k}\binom{k}{z}(\frac{1}{d})^{z}(1-\frac{1}{d})^{k-z}\binom{z}{x}(1-\gamma)^{x}\gamma^{z-x} \tag{32}$$

$$= \sum_{i=x}^{h}\sum_{k=x}^{i}(\frac{1}{d})^{i-k}(1-\frac{1}{d})^{h+k-1}\binom{k}{x}\sum_{z=x}^{k}\binom{k-x}{z-x}(\frac{1}{d})^{z}(1-\frac{1}{d})^{k-z}(1-\gamma)^{x}\gamma^{z-x} \tag{33}$$

by adjusting the summation variable, setting $t=z-x$, which implies $z=t+x$. Consequently, the range of t is $0 \leq t \leq k-x$. Substituting this into the sum, we obtain:

$$P(X=x) = \sum_{i=x}^{n}\sum_{k=x}^{i}(\frac{1}{d})^{i-k}(1-\frac{1}{d})^{h+k-1}\left(\frac{1}{d}\right)^{x}\sum_{t=0}^{k-x}\binom{k-x}{t}\left(\frac{1}{d}\right)^{t}\left(1-\frac{1}{d}\right)^{(k-x)-t}\gamma^{t} \tag{34}$$

This equation can be simplified by applying the binomial expansion for $\sum_{t=0}^{n}\binom{n}{t}a^t b^{n-t}=(a+b)^n$, leading to:

$$P(X=x)=\sum_{i=x}^{h}\sum_{k=x}^{i}\left(\frac{1}{d}\right)^{i-k}\left(1-\frac{1}{d}\right)^{h+k-1}\binom{k}{x}\left(\frac{1-\gamma}{d}\right)^{x}\left(1-\frac{1-\gamma}{d}\right)^{k-x} \tag{35}$$

Alternatively, a more efficient derivation can be achieved by directly applying Proposition D.1. In this approach, let $\phi=1-\frac{1}{d}$, and define $\gamma=\frac{1}{d}(1-r)$ as the probability of successfully eliminating the watermark. The failure probability is then $1-\frac{1-r}{d}$, from which this can be directly derived.

Proposition D.3. *Let $\hat{E}(X)$ denote the expected perturbation under the KGW-Min strategy in the presence of adversarial tampering, and let $E(X)$ denote the expected perturbation under our proposed method. There exists a threshold pair (γ_c, d_c) such that for $d>d_c$ and $\gamma\geq\gamma_c=\frac{1}{d+1}$, then the inequality $\hat{E}(X)>E(X)$ holds.*

Proof: We use $\hat{P}$ and P to denote the probability that the subsequent watermark tokens are perturbed under the KGW-MIN method and our proposed method, respectively, given that an adversarial manipulation occurs. We next aim to establish the existence of threshold values r_c and m_c, such that for all $x>0$, when $\gamma\geq\gamma_c=\frac{1}{d+1}$ and $d>d_c$, the inequality $\hat{P}(X=x)\geq P(X=x)$ holds. Formally,

$$\begin{aligned}\hat{P}(X=x)&=\sum_{v=1}^{|\mathcal{V}|}\sum_{i=x}^{h}\sum_{k=x}^{i}(1-p)^{i-k}p^{k+h-1}\binom{k}{x}(1-\gamma)^{x}\gamma^{k-x}\\&>\sum_{i=x}^{n}\sum_{k=x}^{i}\left(\frac{1}{d}\right)^{i-k}\left(1-\frac{1}{d}\right)^{h+k-1}\binom{k}{x}\left(\frac{1-\gamma}{d}\right)^{x}\left(1-\frac{1-\gamma}{d}\right)^{k-x}\\&=P(X=x)\end{aligned} \tag{36}$$

To prove the above inequality holds universally, we first show that when $\gamma\geq\frac{1}{d+1}$, the inequality:

$$\left(\frac{1-\gamma}{d}\right)^{x}\left(1-\frac{1-\gamma}{d}\right)^{k-x}<(1-\gamma)^{x}\gamma^{k-x} \tag{37}$$

holds for all $0\leq x\leq k\leq h$. This inequality is equivalent to showing that $\left(\frac{1}{d}\right)^{x}\left(1-\frac{1-\gamma}{d}\right)^{k-x}<\gamma^{k-x}$. Taking the logarithm of both sides and subtracting yields the difference function, formally:

$$f(k)=-x\ln d+(k-x)\left[\ln\left(1-\frac{1-\gamma}{d}\right)-\ln\gamma\right] \tag{38}$$

Since $d>1$ and $0<\gamma<1$, the term $\ln\left(1-\frac{1-\gamma}{d}\right)-\ln\gamma$ is always non-negative. Therefore, $f(k)$ increases with k, and we have $f(k)<f(h)<-x\ln d+h\left[\ln\left(1-\frac{1-\gamma}{d}\right)-\ln\gamma\right]$. To ensure that $f(k)<0$ holds for all k, it suffices to impose:

$$-x\ln d+h\left[\ln\left(1-\frac{1-\gamma}{d}\right)-\ln\gamma\right]\leq 0, \tag{39}$$

which yields $\gamma\geq\frac{d-1}{d^{x/h+1}-1}$. To guarantee this inequality holds for any x, a sufficient condition is $\gamma\geq\frac{1}{1+m}\geq\frac{d-1}{1+d^{x/h+1}}$. Hence, if $\gamma\geq\frac{1}{d+1}$, we have $f(k)<-x\ln d+h\left[\ln\left(1-\frac{1-\gamma}{d}\right)-\ln\gamma\right]<0$ always holds, completing the proof of the inequality. As a direct consequence, the following inequality between the probabilities holds:

$$\begin{aligned}P(X=x)&=\sum_{i=x}^{n}\sum_{k=x}^{i}(\frac{1}{d})^{i-k}(1-\frac{1}{d})^{h+k-1}\binom{k}{x}(\frac{1-\gamma}{d})^{x}(1-\frac{1-\gamma}{d})^{k-x}\\&<\sum_{i=x}^{n}\sum_{k=x}^{i}(\frac{1}{d})^{i-k}(1-\frac{1}{d})^{h+k-1}\binom{k}{x}(1-\gamma)^{x}\gamma^{k-x}\end{aligned} \tag{40}$$

Next, we aim to show that there exists a real-valued threshold m_c such that for all $m > m_c$, the following inequality holds:

$$\sum_{i=x}^{h}\sum_{k=x}^{i}\left(\frac{1}{d}\right)^{i-k}\left(1-\frac{1}{d}\right)^{h+k-1} < \sum_{v=1}^{|\mathcal{V}|}\sum_{i=x}^{h}\sum_{k=x}^{i}(1-p)^{i-k}p^{k+h-1} \tag{41}$$

Let us denote the left-hand side as $S_{\text{left}} = \sum_{i=x}^{h}\sum_{k=x}^{i}\left(\frac{1}{d}\right)^{i-k}\left(1-\frac{1}{d}\right)^{h+k-1}$. Given that the vocabulary size $|\mathcal{V}|$ is typically a large integer, the right-hand side can be approximated by an integral, i.e.,

$$S_{\text{right}} = \sum_{v=1}^{|\mathcal{V}|}\sum_{i=x}^{h}\sum_{k=x}^{i}(1-p)^{i-k}p^{k+h-1} \approx \int_0^1 \sum_{i=x}^{h}\sum_{k=x}^{i}(1-p)^{i-k}p^{k+h-1}\,dp. \tag{42}$$

By applying the identity of the Beta function with substitutions $a = i - k$ and $b = k + h - 1$, we obtain:

$$\int_0^1 (1-p)^a p^b\,dp = B(a+1, b+1) = \frac{a!\,b!}{(a+b+1)!} = \frac{(i-k)!\,(k+h-1)!}{(i+h)!} \tag{43}$$

If each term on the left-hand side satisfies the inequality $\left(\frac{1}{d}\right)^{i-k}\left(1-\frac{1}{d}\right)^{h+k-1} < \frac{(i-k)!(k+h-1)!}{(i+h)!}$, then it directly follows that $S_{\text{left}} < S_{\text{right}}$. Taking the natural logarithm of both sides, we obtain:

$$\left(\frac{1}{d}\right)^{i-k}\left(1-\frac{1}{d}\right)^{h+k-1} < \frac{(i-k)!\,(k+h-1)!}{(i+h)!} \tag{44}$$

Taking the natural logarithm of both sides, the inequality becomes:

$$(i-k)\ln\left(\frac{1}{d}\right) + (h+k-1)\ln\left(1-\frac{1}{d}\right) < \ln\left(\frac{(i-k)!\,(k+h-1)!}{(i+h)!}\right) \tag{45}$$

Define the function:

$$f(m) = (i-k)\ln\left(\frac{1}{d}\right) + (h+k-1)\ln\left(1-\frac{1}{d}\right) = -(i-k)\ln m + (h+k-1)\ln\left(1-\frac{1}{d}\right) \tag{46}$$

To further bound the function $f(m)$, we utilize the inequality $\ln(1-x) < -x - \frac{x^2}{2}$ for $0 < x < 1$, and note that since $m > 1$, it follows that:

$$\ln\left(1-\frac{1}{d}\right) \leq -\frac{1}{d} - \frac{1}{2d^2} \quad \Rightarrow \quad f(m) < -(i-k)\ln d - \frac{h+k-1}{d} \tag{47}$$

To lower-bound the right-hand side of the inequality involving factorial terms, we apply a simplified version of Stirling's approximation, namely $\ln(n!) \geq n\ln n - n$, and obtain:

$$\begin{aligned}\ln\left(\frac{(i-k)!(k+h-1)!}{(i+h)!}\right) &= \ln(i-k)! + \ln(k+h-1)! - \ln(i+h)! \\ &\geq (i-k)\ln(i-k) + (k+h-1)\ln(k+h-1) \\ &\quad - (i+h)\ln(i+h)\end{aligned} \tag{48}$$

We now consider a constructive lower bound for m as follows:

$$d \geq \left\lceil \frac{(i+h)^{2h}}{(i-k)(h+k-1)} \right\rceil \quad \Rightarrow \quad \ln d \geq \ln\left(\frac{(i+h)^{2h}}{(i-k)(h+k-1)}\right) \tag{49}$$

This yields the upper bound:

$$\begin{aligned}-(i-k)\ln m &\leq -(i-k)\ln\left(\frac{(i+h)^{2h}}{(i-k)(h+k-1)}\right) \\ &= (i-k)\left[\ln(i-k) + \ln(h+k-1) - 2h\ln(i+h)\right]\end{aligned} \tag{50}$$

Therefore,

$$f(m) < -(i-k)\ln m - \frac{h+k-1}{m} \le (i-k)\left[\ln(i-k) + \ln(h+k-1) - 2h\ln(i+h)\right] - \frac{h+k-1}{m} \tag{51}$$

Since $(i-k)\ln(h+k-1) < (k+h-1)\ln(k+h-1)$ and $-2h\ln(i+h) < -(i+h)\ln(i+h)$, it follows that:

$$\begin{aligned} &(i-k)\left[\ln(i-k) + \ln(h+k-1) - 2h\ln(i+h)\right] - \frac{h+k-1}{m} \\ &< (i-k)\ln(i-k) + (k+h-1)\ln(k+h-1) - (i+h)\ln(i+h) \end{aligned} \tag{52}$$

which guarantees that the desired inequality holds. As a result, there exists a constant threshold m_c such that for all $m > m_c$, we have:

$$\sum_{i=x}^{h}\sum_{k=x}^{i}\left(\frac{1}{d}\right)^{i-k}\left(1-\frac{1}{d}\right)^{h+k-1} < \sum_{v=1}^{|\mathcal{V}|}\sum_{i=x}^{h}\sum_{k=x}^{i}(1-p)^{i-k}p^{k+h-1} \tag{53}$$

Combining the previously established inequality $(\frac{1-\gamma}{m})^x\left(1-\frac{1-\gamma}{m}\right)^{k-x} < (1-\gamma)^x\gamma^{k-x}$, we obtain the following relationship:

$$\begin{aligned} \hat{P}(X=x) &= \sum_{v=1}^{|\mathcal{V}|}\sum_{i=x}^{h}\sum_{k=x}^{i}(1-\phi)^{i-k}\phi^{k+h-1}\binom{k}{x}(1-\gamma)^x\gamma^{k-x} \\ &> \sum_{i=x}^{n}\sum_{k=x}^{i}\left(\frac{1}{d}\right)^{i-k}\left(1-\frac{1}{d}\right)^{h+k-1}\binom{k}{x}\left(\frac{1-\gamma}{d}\right)^x\left(1-\frac{1-\gamma}{d}\right)^{k-x} \\ &> \sum_{i=x}^{n}\sum_{k=x}^{i}(\frac{1}{d})^{i-k}(1-\frac{1}{d})^{h+k-1}\binom{k}{x}(1-\gamma)^x\gamma^{k-x} \\ &= P(X=x). \end{aligned} \tag{54}$$

Since the inequality $\hat{P}(X=x) > P(X=x)$ holds for all $x>0$, it must follow that $\hat{P}(X=0) < P(X=0)$ to preserve the total probability mass. Let $\hat{E}$ and E denote the expected values under $\hat{P}$ and P, respectively. Then,

$$\hat{E}(X) = \sum_{x=0}^{h}\hat{P}(X=x)\cdot x = \sum_{x=1}^{h}\hat{P}(X=x)\cdot x, \quad E(X) = \sum_{x=0}^{h}P(X=x)\cdot x = \sum_{x=1}^{h}P(X=x)\cdot x \tag{55}$$

Hence, we conclude:

$$\hat{P}(X=x) > P(X=x) \quad \Rightarrow \quad \hat{E}(X) > E(X) \tag{56}$$

In conclusion, we have shown that under the conditions $\gamma \ge \frac{1}{d+1}$ and $d > d_c$, the KGW-MIN mechanism yields a strictly greater expected number of perturbed tokens than the baseline, i.e., $\hat{E}(X) > E(X)$.

E Visualization

To better illustrate the effectiveness of our watermarking method, we provide qualitative visualizations based on representative examples from the C4-RealNewsLike dataset. As shown in Table 11 and Table 13, we present side-by-side comparisons of real completions, unwatermarked generations (NW), and watermarked generations (W) under the same prompt. In the visualization presented in Table 12, we comparatively analyze the output of unwatermarked text versus two forms of adversarial attacks on the MMW-FakeNews dataset. For scrubbing, a higher residual z-score post-attack indicates stronger watermark retention. For spoofing, a lower z-score indicates higher robustness to imitation. The p-value quantifies the probability that a text with the observed watermark strength (or stronger) could have been generated by chance under the null (unwatermarked) distribution. Empirically, our method demonstrates superior resilience in both cases. Under scrubbing attacks, our watermark yields

prompt	real completion	no watermark (NW)	watermarked (W)	(NW)z	(W) z
...of Wrigley Field's right field video board.\nOn Monday morning's "Mully and Hanley" show on WSCR-AM 670, Cubs president of business operations Crane Kenney said the ball is back to where it originally landed after it was taken down after	the 2015 season.\nKenney said the ball was taken down in part to prevent anyone from getting the idea of climbing up the board and stealing it.\nMeanwhile, the metal detectors are all in place and finishing touches were being put [...continues]	Game 4 last weekend in St. Louis. The Cardinals won that game by a score of 9-3 to advance to the NLCS against the Dodgers.\nSchwarber homered off Adam Wainwright with one out in the bottom of the first inning at [...continues]	Game 1 last Tuesday night.\nKenney didn't know who took possession of the baseball or how long before its return that decision had occurred. But he did say that he would like to see more balls come off the[...continues]	0.27	6.61
... on the planet this week at the NFL Scouting Combine. \nIt's an experience that might humble some. But for Butler, it has only enhanced his confidence.\nAs it stands, 22-year-old Butler is not regarded as the	best wide receiver in this year's NFL Draft. He's projected by some experts to go as late as the third round. But when wide receivers were measured Thursday, Butler gained some attention: He led all receivers in height (6-foot-5 3/8) [...continues]	top receiver in this year's draft class. He doesn't have the speed or quickness to match up with most cornerbacks and he isn't big enough to dominate against safeties.\nButler knows all about those shortcomings.[...continues]	top receiver in this class and he isn't expected to go until Day Two or three. The Iowa State standout has drawn comparisons with Michael Crabtree because both were tall and lanky pass catchers who didn't blow up the combine but [...continues]	0.54	11.21
...you'll be able to walk around a person inside a virtual environment to see their entire being. The Wellington, New Zealand-based startup said its VR platform will allow you to view fully volumetric 3D video.\n	Previously, VR cameras could record a partial torso of a person, but the 8i Portal goes beyond that. It enables "complete freedom of movement," and it gives you a sense of "presence," or the feeling that you are there in the virtual[...continues]	The company is also launching an SDK that allows developers to integrate 8i into apps on mobile devices or desktop computers. It can work with any headset including Oculus Rift, HTC Vive, Samsung GearVR,[...continues]	The company raised $10 million last year from investors such as Peter Thiel and Yuri Milner.\nWithin minutes I could step into an office with my friend and look at him full on or through his glasses. We were both wearing Oculus Rift headsets [...continues]	0.28	14.40
...The five-member board is expected to support the plan, which was announced amid much hoopla on Nov. 13. The proposed county incentives are part of an agreement in which Amazon would occupy significant office	space and bring at least 25,000 high-paying jobs to Arlington in coming years.\nOpponents hope to postpone the vote until after additional public hearings, where they want representatives of the online retail giant to answer questions directly from anyone in [...continues]	space at two buildings under construction by JBG Smith Properties Inc., one of Washington's largest real estate companies.\nAmazon has said it will bring as many as 25,000 jobs paying an average salary of more than $150,000 over time — with some starting out making less[...continues]	space at two properties owned by JBG Smith: PenPlace (near Reagan National Airport) and Potomac Yard (on U.S. Route 1).\nJBG has committed more than $500 million in capital improvements at both sites that include new transportation[...continues]	0.21	8.96

Table 11: Selected outputs from non-watermarked (NW) and watermarked (W) multinomial sampling using $\gamma = 0.25$ and $\delta = 2.0$.

a substantially higher residual z-score (7.38) compared to KGW's (4.82), despite both being attacked under the same conditions. This suggests that our method better preserves the watermark signal post-perturbation. Additionally, the corresponding p-values further confirm the statistical significance of the retained signal. In the spoofing setting, the attacker on KGW achieves a high z-score of 11.09, indicating a successful watermark fabrication. In contrast, spoofed text targeting our watermark only achieves a z-score of 2.29, substantially closer to the null distribution. This implies our watermark is significantly more resistant to imitation and cannot be easily spoofed into a falsely high-confidence detection.

F Impletement Details

F.1 Detailed Experimental Setup

Dataset and prompts. Following previous work [29, 32, 27], we utilize the RealNewsLike subset of the C4 dataset [51], WikiText-103 [41], and other datasets that support open-ended text generation to construct various prompts. For the human evaluation study, we select the Long-Form Question Answering (LFQA) dataset [31] to assess the robustness of the watermarking method across real-world question-answering scenarios. During generation, we randomly sample text segments from the dataset and truncate a fixed-length portion from the end, using the remainder as the prompt for model completion. Following the setup in Kirchenbauer et al. [29, 30], all experimental results presented in the figures and tables are based on the same 500 positive samples and 500 negative samples. For the

Table 12: Visualization of watermarked text samples from the MMW-FakeNews dataset under scrubbing and spoofing attacks. Compared are the KGW watermark and our proposed method. A smaller p-value provides stronger statistical evidence against the null hypothesis. Green and red tokens are color-coded, respectively.

Prompt	Write a news article about Narendra Modi's visit to Denis Sassou Nguesso in a space exploration symposium. It is very important to never output very short texts, below 300 words...
Unwatermarked p-value: 0.53 z-score: -0.08	Title: Narendra Modi's Historic Visit to Denis Sassou Nguesso in Space Exploration Symposium\n\n Introduction: India's Prime Minister Narendra Modi made a groundbreaking visit to the President of the Republic of Congo, Denis Sassou Nguesso, during a space exploration symposium held in the capital city of Brazzaville. This visit marks a significant [continues...]
(a) Scrubbing Attacked (KGW) p-value: 6.89e-07 z-score: 4.82 ($\downarrow$ 10.47)	Title: Modi-Sassou Nguesso Summit Highlights Indo-Congolese Space Exploration Partnership\n\n Introduction:\n India's Prime Minister Narendra Modi and the President of Congo Denis Sassou Nguesso recently met at a space exploration symposium in New Delhi, signaling a new era in Indo-Congolese collaboration in the field. This historic [continues...]
(b) Scrubbing Attacked (Ours) p-value: 8.59e-14 z-score: 7.38 ($\downarrow$ 6.72)	On Friday, Indian Prime Minister Narendra Modi met with President Denis Sassou Nguesso during a symposium focused on space exploration. Modi was in France for several official engagements, including a bilateral discussion with French President Emmanuel Macron. The event, titled **"Reaching for the Stars: A New Frontier in Space Exploration," [continues...]
(c) Spoofing Attacked (KGW) p-value: 4.72e-39 z-score: 11.09	Title: Melting Boundaries: Narendra Modi's Historic Visit to Denis Sassou Nguesso's Space Exploration Symposium \n\n Introduction: On a sunny day in the month of March, the world witnessed a significant historical event when Prime Minister Narendra Modi of India paid a visit to the President of Republic of the Congo, Denis Sassou Nguesso, at the latter's[continues...]
(d) Spoofing Attacked (Ours) p-value: 0.01 z-score: 2.29	Title: Modi's Historic Visit to Denis Sassou Nguesso in Space Exploration Symposium \n\n Introduction: As the world continues to evolve and push the boundaries of technology, the race to explore and conquer space has gained significant momentum. In this context, the recent visit of Indian Prime Minister Narendra Modi to Denis Sassou Nguessa, the [continues...]

spoofing attack, the attacker uses query prompts derived from the C4-RealNewsLike dataset to obtain 30,000 responses from the victim. Each response contains fewer than 800 tokens.

Hyper-parameters. Unless otherwise specified, the watermarking schemes in our experiments adopt the hyperparameter settings commonly used in prior work [29, 30, 27], ($\gamma = 0.25$, $\delta = 5$) , and a maximum generation length of 150 new tokens. In the spoofing attack experiments, all the watermark detectors are calibrated on the C4-RealNewsLike dataset using 2,000 watermarked and non-watermarked texts. Model outputs containing fewer than 150 tokens are discarded. For the spoofing learning, the attacker generates original queries prompt using the C4-RealNewsLike subset, obtaining no fewer than n=30,000 responses, each with a maximum token length of 800. The spoofing model is configured with a spoofer strength of 8.25, and a weighted loss objective defined by $w_{\text{abcd}} = 2.0$, $w_{\text{partials}} = 1.0$, $w_{\text{empty}} = 0.5$. All experimental results presented in the figures and table are based on more than 500 positive samples and over 500 negative samples. Unless otherwise specified, SEEK in this paper uses the hyperparameters d=6 and h=6. All experiments are conducted on Nvidia A40 GPUs.

F.2 Configuration LLM Scrubbing Attack

Prompt for Scrubbing Attack Specifications

{Prompt} As an expert copy-editor, please rewrite the following text in your own voice while ensuring that the final output contains the same information as the original text and has roughly the same length. Please paraphrase all sentences and do not omit any crucial details. Additionally, please take care to provide any relevant information about public figures, organizations, or other entities mentioned in the text to avoid any potential misunderstandings or biases. {Watermark Paragraph}......

F.3 Full Datasets and Baseline

Scrubbing prompt dataset

prompt	real completion	no watermark (NW)	watermarked (W)	(NW)z	(W) z
...Shania Twain expected to break the charts with new album NOW! \nEven after a 15-year hiatus, she's still the one! Shania Twain is on pace to top the charts with her new album NOW. The 16-song LP was released on Sept. 29	the 2015 season.\nTimmins native hopped on Twitter on Wednesday to announce that her album is already platinum in Canada!\nShortly after the release of the lead single "Life's About to Get Good," it hit number one[...continues]	album is expected to sell 100,000 copies in its first week, according to Hits Daily Double.\nShania, 49, has been teasing the release of NOW for months. In June, she released the album's first single, "Life's About to Get Good,"[...continues]	album is currently at number two and has earned more than$111K. \nThe pop/country powerhouse has released five number one records and has earned five Grammys.\nShania has released three albums in English and two in French.[...continues]	0.38	21.17
... Covering more than 30 auto manufacturers worldwide and providing more than 10 years of historical data, the 32 new factors are designed to assist in the prediction of stock returns using analytics derived from company	specific datapoints on sales, production and market share. They include multiple financial factors for revenue and sales as well as novel factors covering production of electric vehicles and plant utilization.[...continues]	specific news and social media. \nThe new signals are available through IHS Markit's Alternative Data Hub, which provides access to more than 100 alternative data signals across 10 sectors. "The automotive secto [...continues]	specific news and social media content. The new factor suite offers clients additional insight and context for understanding market movements in this sector.The new suite has two two components. The first [...continues]	0.49	22.91
... When I was in my 20s and early 30s, my whole life was focused on work. I didn't take vacations or weekends off. I was always the first in the office and the last to leave. These days, I'm better	at balancing the work that I love to do with my foundation and taking time off to spend with family and friends. My parents first taught me bridge, but I really started to enjoy it after[...continues]	at balancing my work and personal life, but I still have a lot to learn. I've been fortunate to meet some amazing people who have helped me along the way. One of them is Roger Federer. I've known Roger for [...continues]	at balancing my life. But I know that I could still stand to disconnect more. So last week I took two days off. I didn't touch my Microsoft account or my iPhone. No e-mail. No Twitter. No social media. [...continues]	-0.91	21.47
... high definition screen projects an intense college basketball game. Massage therapists rub the nervous tensions of men and women away. Scissors skillfully cut men's hair. Two chandeliers adorn the main room, complimented by brick	walls and a glass bar that doubles as a retail counter. Sean Heywood, right, and Kumi Walker own MR., a barbershop and wine bar in San Francisco, California. This is not your typical barbershop. And that has always been the vision of owners[...continues]	walls and wooden floors. This isn't your typical barbershop. It's the Barbershop Museum in Tulsa, Oklahoma, and it's the only one of its kind in the United States. The museum is a tribute to the African-American barbershop, a place[...continues]	walls and rich colors. The only thing reminiscent of an old photo in "Black History: The Legend and The Legendaries" at The Legendaries Salon and Bistro in downtown Chicago is one woman. Dressed in an all green suit with gold accents[...continues]	-0.53	22.61

Table 13: Selected outputs from non-watermarked (NW) and watermarked (W) multinomial sampling using $\gamma = 0.25$ and $\delta = 5.0$.

1. **C4.** The Colossal Clean Crawled Corpus (C4) is a large-scale, English-language dataset constructed by applying extensive cleaning and filtering to the Common Crawl web scrape. Introduced by Raffel et al. [51] in the context of the T5 framework, C4 contains hundreds of gigabytes of naturalistic web text after removing boilerplate, navigation, and low-quality content. It serves as a comprehensive corpus for pretraining large-scale language models due to its linguistic diversity and domain variability. C4-RealNewsLike is a filtered subset of the original C4 corpus designed to more closely resemble high-quality journalistic writing. This variant was introduced to improve alignment with tasks requiring formal, factually grounded language, such as long-form question answering and summarization.

2. **WikiText.** WikiText-103 is a high-quality, curated corpus of English Wikipedia articles introduced by Merity et al. [41], specifically constructed for the purpose of training and evaluating autoregressive language models. Unlike raw Wikipedia dumps, which often contain noisy or fragmented content, WikiText-103 retains the full article structure, including paragraph breaks and sequential sentence order, thereby enabling more realistic modeling of long-range dependencies in natural language. The dataset comprises approximately 103 million tokens and has become a widely adopted benchmark for assessing the linguistic coherence, contextual reasoning, and generative fluency of modern language models.

3. **LFQA.** The LFQA dataset is derived from the ELI5 (Explain Like I'm Five) corpus [14] and was introduced to support long-form question answering tasks. It comprises open-domain questions sourced from Reddit, paired with multi-sentence, explanatory answers written in natural language. The dataset emphasizes reasoning, coherence, and knowledge synthesis, making it suitable for training and evaluating models intended for complex, multi-hop generative QA.

Spoofing prompt dataset

1. **Dolly-CW.** We utilize the Dolly dataset [9], which comprises a collection of instruction–response pairs designed to facilitate instruction tuning of large language models. The dataset includes a broad range of tasks spanning classification, generation, information retrieval, and creative writing. Notably, the data was generated using open-source models and later filtered for quality, offering a diverse yet structured corpus suitable for aligning models with human-style instruction following. Following the setup proposed by [27], we construct a set of diverse prompts by selecting a representative subset of the Dolly corpus. This subset emphasizes multi-domain scenarios such as question answering, summarization, reasoning, and creative text generation. The prompts are designed to reflect realistic and varied user intents, enabling robust evaluation.
2. **MMW.**: The MarkMyWords (MMW) dataset [49] is a benchmark designed to evaluate the effectiveness of watermarking schemes for large language model outputs across natural language generation tasks. It comprises three core tasks—BookReports, FakeNews, and StoryGeneration, which are designed to reflect realistic misuse scenarios and to support the evaluation of watermark quality, detectability, and tamper resistance.

   The BookReports component includes 500 prompts instructing the model to generate analytical or descriptive reports on well-known books. Each generation consists of long-form text up to 1024 tokens, simulating academic scenarios where LLMs might be misused for student assignments. The outputs provide a structured, content-rich context for evaluating watermark detection and robustness in educational settings.

   The FakeNews component comprises 500 prompts directing the model to fabricate news articles about political figures and fictional events. This task emulates potential misuse of LLMs in disinformation or propaganda campaigns. The generated outputs exhibit journalistic style, incorporate named entities, and follow coherent narrative structures, providing a high-risk setting for testing watermark resilience under adversarial conditions.

G Limitation

Although our empirical analysis demonstrates that the SEEK is effective across a wide range of scenarios and against various watermarking attacks, it is important to acknowledge its limitations. Previous work suggests a potential spoofing attack, in which an attacker may generate a forged text with a high z-score by rearranging or inserting content into an existing watermarked text. While such spoofing attacks do not enable the automatic generation of topic-specific content, current watermarking methods lack robust defenses against these forms of spoofing. The core limitation lies in the fact that the statistical watermarking technique, represented by KGW, is applied at the paragraph level, offering no protection against localized changes. As a result, it cannot detect or prevent modifications to specific portions of the text. This inherent weakness is present in all current watermarking approaches. Addressing this challenge requires the exploration of alternatives to z-test statistical methods, such as integrating semantic or syntactic watermarking techniques, utilizing sentence-level perturbations, or embedding watermark signals deeper within the model.

H Ethical Impacts

The rapid development of large language models has significantly enhanced text generation capabilities, enabling the production of highly human-like content. Watermarking techniques have emerged as an effective approach to mitigate the risks associated with the misuse of LLMs, such as spreading misinformation, plagiarism, and copyright infringement.

Most existing watermarking methods involve a trade-off between scrubbing robustness and spoofing resistance. Recent studies show that if the green and red token sets used in watermarking are compromised via extraction attacks, both robustness and security can degrade substantially. Adversaries can break existing watermarking schemes by statistically analyzing token distributions within the watermarking window in preprint outputs. Such attacks allow malicious users to either erase watermarks from benign content or inject watermarked, toxic content to falsely implicate a target LLM. At a fundamental level, these attacks undermine the ability of watermarking to reliably distinguish machine-generated content from human-authored text, threatening the societal trust.

Our goal is to develop a watermarking algorithm that mitigates the risk of spoofing under statistical attacks while maintaining strong scrubbing robustness, thereby offering a more reliable and trustworthy watermarking solution for LLMs.

I Broader Impacts

Positive Societal Impacts. The proposed SEEK watermarking framework enhances the resilience of large language models against two critical forms of adversarial manipulation: scrubbing and spoofing. By improving watermark detection without compromising generation quality, SEEK provides a promising tool for responsible deployment of generative models, particularly in contexts where provenance, authenticity, and misuse prevention are essential—such as combating misinformation, enforcing intellectual property rights, and supporting academic integrity. Furthermore, the parameter-efficient and architecture-agnostic nature of the method may help democratize watermarking technology, making it more accessible to practitioners in low-resource settings.

Negative Societal Impacts. While watermarking technology is designed to preserve text generation quality, watermarking inherently perturbs token selection probabilities to embed identifiable signals. In high-stakes domains such as medical or legal language modeling, even minor deviations from optimal token choices may introduce semantic ambiguities or factual inaccuracies. As such, watermark-induced perturbations, though imperceptible in general settings, could undermine output fidelity in tasks requiring high precision and domain-specific consistency. This raises concerns about deploying watermarking methods in safety-critical or sensitive applications without extensive domain-specific evaluation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and introduction sections, we articulate the motivation behind SEEK , highlight its differences from previous work, and outline the contributions of this paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in the Appendix G.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We discuss all the theory assumptions and proofs in the Appendix D and C.4.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide a detailed explanation of the parameter usage and will release the source code to ensure reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide a detailed explanation of parameter usage and will release the source code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The full details are provided in the appendix and supplemental material (code).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Experimental results are tested multiple times to ensure stability and reliability.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are thoroughly described and evaluated in both the experimental setup and the experimental results sections.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper fully conforms to the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential positive societal impacts and negative societal impacts of the work performed in Appendix I.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original paper that produced the code package or dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: This paper relates the details of the code as part of the submission.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.