WaterPool: A Language Model Watermark Mitigating Trade-Offs among Imperceptibility, Efficacy and Robustness

Baizhou Huang [♣♡] **Xiaojun Wan** [♣♡]
[♣]Wangxuan Institute of Computer Technology, Peking University
[♡]State Key Laboratory of General Artificial Intelligence
{hbz19,wanxiaojun}@pku.edu.cn

Abstract

Watermarking is a prominent technique to trace the usage of specific large language models (LLMs) by injecting patterns into model-generated content. An ideal watermark should be imperceptible, easily detectable, and robust to text alterations, yet existing methods typically face trade-offs among these properties. This paper utilizes a key-centered scheme to unify existing methods by decomposing a watermark into two components: a key module and a mark module. We show that the trade-off issue is the reflection of the conflict between the scale of the key sampling space during generation and the complexity of key restoration during detection within the key module. To this end, we introduce **WaterPool**, a simple yet effective key module that preserves a complete key sampling space for imperceptibility while utilizing semantics-based search to improve the key restoration process. WaterPool can integrate seamlessly with existing watermarking techniques, significantly enhancing their performance, achieving near-optimal imperceptibility, and markedly improving their detection efficacy and robustness (+12.73% for KGW, +20.27% for EXP, +7.27% for ITS)[1].

1 Introduction

The world has recently witnessed the great power of large language models (LLMs). However, the widespread use of these systems has raised significant concerns about their potential misuse. For example, LLMs could be used to generate massive amounts of fake news or automated comments to manipulate social media, posing threats to academic integrity and intellectual property rights (Bender et al., 2021; Liu et al., 2023b).

To address these issues, watermarking has been proposed to track the usage of specific models (Kirchenbauer et al., 2023a). It embeds a hidden pattern into generated contents of a specific LLM during decoding, which is conduct by sampling outputs from a stochastic modified distribution instead of the original language modeling distribution. Ideally, the expectation of the modified distribution is nearly identical to the original one, making the watermarked text almost indistinguishable from the original (**imperceptibility**). This pattern can be reliably detected by a detector (**efficacy**) and remains high detection rate even if the text is corrupted by semantic-preserving attacks (**robustness**).

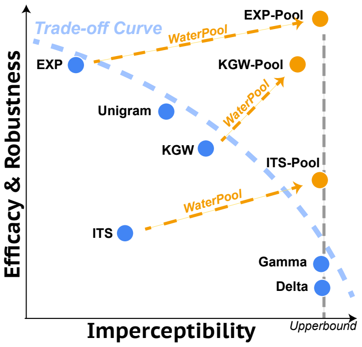

Figure 1: Previous methods face trade-offs among imperceptibility, efficacy and robustness. WaterPool mitigates this problem to a significant extent. It can be integrated with other watermarking methods, improving them on all three aspects. This figure is based on the experiments on open-ended text generation presented in Table 1 and 2.

Prior works have made great progress towards these properties for an ideal watermark (Kirchenbauer et al., 2023a,b; Kuditipudi et al., 2023; Zhao et al., 2023; Hu et al., 2023). However, none of them have achieved all three properties simultaneously as illustrated in Figure 1. It is widely accepted that there is a trade-off among imperceptibility, efficacy, and robustness. Previous methods often use hyper-parameters to balance this trade-off issue, like the δ in KGW controlling the degree of distribution shift.

In this paper, we try to explore this issue. We

[1] The code is available in https://github.com/skpig/waterpool.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 4156–4182
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

begin by unifying existing watermarking methods with a key-centered scheme. It decomposes a watermarking technique into two independent modules, a *key module* and a *mark module*, as shown in Figure 2. During generation, the key module samples a private key. It is then utilized by the mark module as a random seed to distort the next token distribution, from which watermarked texts are sampled. During detection, the key module attempts to restore the private key. Then the mark module aligns the text with the restored key to compute statistics, which imply the likelihood of watermark presence. The decomposition separates requirements of imperceptibility, efficacy and robustness into the two modules. Subsequently, we show that the key module significantly contributes to the trade-off problem. Specifically, the trade-off problem actually stems from the conflict between the scale of key sampling space during generation and the complexity of key restoration during detection.

To overcome this trade-off, we introduce **WaterPool**, a simple but effective key module. WaterPool maintains the complete key sampling space, crucial for imperceptibility, while leveraging a semantics-based search to significantly enhance the precision and effectiveness of the key restoration process, thereby ensuring high robustness against attacks. We integrate WaterPool into three of the most renowned watermarking techniques, EXP (Kuditipudi et al., 2023)), KGW (Kirchenbauer et al., 2023a) and ITS (Kuditipudi et al., 2023)). WaterPool effectively mitigates the traditionally "inevitable" trade-offs, achieving superior performance as shown in Figure 1.

Our experiments include two scale of large language models (LLMs) across tasks of open-ended generation and long-form question answering. Experimental results demonstrate the supreme capabilities of our proposed WaterPool. On one hand, it elevates the imperceptibility of KGW, EXP and ITS to near-optimal levels. On the other hand, it significantly enhances the efficacy and robustness of previous watermarking techniques, yielding substantial improvements across different experimental settings (+12.73% for KGW, +20.27% for EXP, +7.27% for ITS)[2].

2 Preliminary

Problem Formulation. We begin by formalizing the process of watermarking. Given any prompt $\mathbf{x}$, a LLM will generate a sequence of output token $\mathbf{y}_i \sim P_M(\cdot|\mathbf{x}, \mathbf{y}_{<i})$ in an auto-regressive manner. A watermark will stochastically distort the distribution to a *Modified Distribution* $\hat{P}_M(\cdot|\mathbf{x}, \mathbf{y}_{<i}) \in \Delta(\Sigma)$ over the vocabulary Σ. The detection of watermarked texts is formulated as a hypothesis testing problem with an alternative hypothesis that the candidate is sampled from a modified distribution. It is typically proved by gathering per-token statistics s_i for significance test.

Imperceptibility. An ideal watermark should maintain the output distribution of LLM as unchanged as possible (Christ et al., 2023; Hu et al., 2023; Kuditipudi et al., 2023; Liu et al., 2023b). Formally, we define the imperceptibility following "N-shot undetectable" from Hu et al. (2023): for all $\mathbf{x}^n, \mathbf{y}^n \in \Sigma^*$,

$$\prod_n^N P_M(\mathbf{y}^n|\mathbf{x}^n) = \mathbb{E}_{\hat{P}_M}[\prod_n^N \hat{P}_M(\mathbf{y}^n|\mathbf{x}^n)] \tag{1}$$

Here, superscripts n indicates different rounds of generation. For LLMs, which model language in an auto-regressive manner, the equation above can be expressed as:

$$\prod_{i,n} P_M(\mathbf{y}_i|\mathbf{x}^n, \mathbf{y}^n_{<i}) = \mathbb{E}_{\hat{P}_M}[\prod_{i,n} \hat{P}_M(\mathbf{y}_i|\mathbf{x}^n, \mathbf{y}^n_{<i})] \tag{2}$$

We want to highlight the importance of the product over multiple generations above. It indicates that it is infeasible to distinguish between the original and the watermarked texts without prior knowledge about the modified distribution, even when multiple queries are allowed (Christ et al., 2023).

Efficacy. An ideal watermark technique should be able to distinguish watermarked texts from the others. Empirically, it is required to achieve high true positive rate with low false positive rate. Most watermarking techniques achieve this by ensuring a substantial difference between the per-token statistic under alternative hypothesis (H_1: the candidate token is sampled from the modified distribution $\hat{P}_M(\cdot|\mathbf{x}^n, \mathbf{y}^n_{<i})$) and null hypothesis ($H_0$: the candidate token is sampled from other distributions). This can be formulated as,

$$\mathbb{E}[s_i|H_1] - \mathbb{E}[s_i|H_0] \geq \phi(\boldsymbol{p}^i), \tag{3}$$

where s_i is the statistic of i-th token. We denote the left-hand side as *Statistical Difference*. $\phi(\boldsymbol{p}^i)$ only depends on $\boldsymbol{p}^i$, the probability vector for i-th step distribution $P_M(\cdot|\mathbf{x}, \mathbf{y}_{<i})$, remaining constant given $\mathbf{x}$ and $\mathbf{y}_{<i}$. It indicates the innate potential

[2] The results are the performance of WaterPool with OPT-1.3B on the open-ended generation task

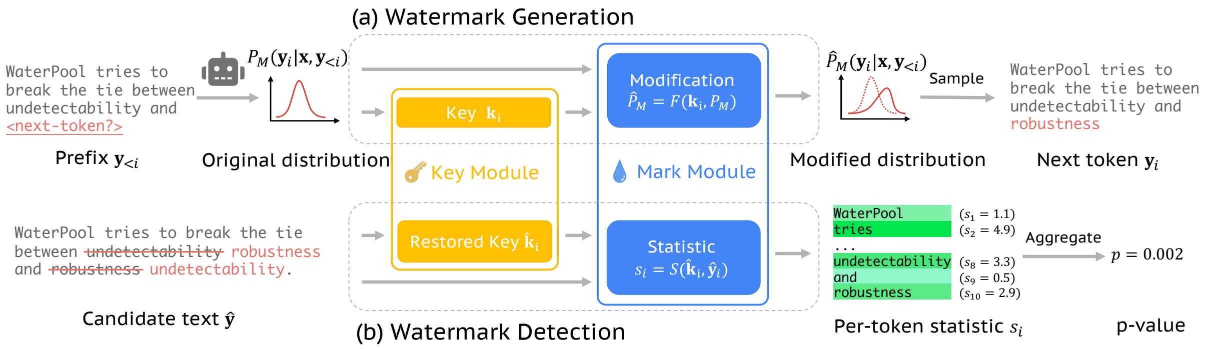
(a) Watermark Generation
WaterPool tries to break the tie between undetectability and <next-token?>
Prefix $\mathbf{y}_{<i}$
$P_M(\mathbf{y}_i|\mathbf{x}, \mathbf{y}_{<i})$
Original distribution
Key $\mathbf{k}_i$
Key Module
Modification
$\hat{P}_M = F(\mathbf{k}_i, P_M)$
Mark Module
$\hat{P}_M(\mathbf{y}_i|\mathbf{x}, \mathbf{y}_{<i})$
Sample
Modified distribution
WaterPool tries to break the tie between undetectability and robustness
Next token $\mathbf{y}_i$
WaterPool tries to break the tie between ~~undetectability~~ robustness and ~~robustness~~ undetectability.
Candidate text $\hat{\mathbf{y}}$
Restored Key $\hat{\mathbf{k}}_i$
Statistic
$s_i = S(\hat{\mathbf{k}}_i, \hat{\mathbf{y}}_i)$
WaterPool ($s_1 = 1.1$)
tries ($s_2 = 4.9$)
...
undetectability ($s_8 = 3.3$)
and ($s_9 = 0.5$)
robustness ($s_{10} = 2.9$)
Aggregate
$p = 0.002$
Per-token statistic s_i
p-value
(b) Watermark Detection

Figure 2: Overview of key-centered watermarking scheme. A watermark is decomposed into two modules, a key module and a mark module. (a) During generation, the LLM provides an next token distribution P_M. The key module samples a private key $\mathbf{k}_i$ as a random seed for the mark module to stochastically modify the distribution to $\hat{P}_M$, from which watermarked texts are sampled. (b) During detection, the key module restores the key $\hat{\mathbf{k}}_i$ for each candidate token. The mark module then calculates the per-token statistic s_i based on the restored key and aggregates them for p-value.

for watermark injection[3]. The statistical difference enables the application of statistical tests like permutation tests (Kuditipudi et al., 2023) or parametric tests assuming specific distribution forms under the null hypothesis (Kirchenbauer et al., 2023a; Fernandez et al., 2023).

Robustness. Robustness is a one-step-further requirement of efficacy. The statistical pattern of watermarked texts could be vulnerable to potential attacks, including lexical modification or paraphrasing (Krishna et al., 2023; Kirchenbauer et al., 2023b). An ideal watermark technique should be resilient to removal and should maintain high efficacy even after such semantics-preserved attacks.

3 Methods

3.1 Decompose Watermark: a Key-centered Scheme

As stated in Section 2, the critical part of watermarking falls in the modified distribution, which is stochastic and determined by a random seed, i.e. the *Private Key* (Christ et al., 2023; Kuditipudi et al., 2023). With private keys as connection, we decompose watermarks into two independent modules: a *Key Module* and a *Mark Module*. The former handles the sampling and restoration of private keys, while the latter is responsible for the modification process and per-token statistic based on private keys. The overall scheme is illustrated in Figure 2.

Specifically, during the i-th step of generation, the key module samples a private key $\mathbf{k}_i$ from the possible key space $\Xi \subset \mathbb{R}$ to provide randomness. Then the mark module takes the sampled key as a random seed to stochastically modify the next token distribution $P_i := P_M(\cdot|\mathbf{x}^n, \mathbf{y}^n_{<i})$ to $F(\mathbf{k}_i, P_i)$, where $F : \mathbb{R} \times \Delta(\Sigma) \to \Delta(\Sigma)$.

During detection, a candidate text $\hat{\mathbf{y}}$ is given. Watermarking techniques generally frame the detection as a hypothesis testing problem, treating each token as an i.i.d sample. For each token $\hat{\mathbf{y}}_i$, the key module tries to restore the corresponding private key $\hat{\mathbf{k}}_i$ used in generation based on the context. The mark module then calculates the per-token statistic $s_i = S(\hat{\mathbf{y}}_i, \hat{\mathbf{k}}_i)$, where $S : \mathbb{R} \times \Sigma \to \mathbb{R}$. These statistics are then aggregated to indicate the likelihood of the entire sequence being watermarked.

In this scheme, the key module and the mark module operate independently, which allows for the combination of any key module with any mark module to create new watermarking methods (Piet et al., 2023). We review several well-known watermarking techniques, list their designs of mark modules (i.e. F and S) and key modules in Appendix A. In specific, we mainly focus on three renowned watermarking techniques in this work:

- KGW (Kirchenbauer et al., 2023a). For the key module, KGW utilizes the hash value of the c-length context $\mathbf{y}_{i-c:i-1}$ as the private key $\mathbf{k}_i$. For the mark module, it randomly samples a partition of the vocabulary Σ, denoted as the green list $\mathcal{G}_{\mathbf{k}_i}$ [4]. The logits of green list

[3] $\phi(\boldsymbol{p})$ has different forms and names in previous works. (Kuditipudi et al., 2023) defines it as watermark potentials. (Kirchenbauer et al., 2023a) connects it with a special form of entropy.

[4] Thoughout the paper, we use the subscript $_{\mathbf{k}_i}$ to indicate a random variable seeded by the private key $\mathbf{k}_i$.

tokens are added by a constant τ to form the modified distribution.

- EXP (Kuditipudi et al., 2023). For the key module, EXP limits the possible key space Ξ to a finite set sampled from $\mathbb{R}$. With the finite key space, the key restoration process during detection can be estimated via greedy search. Specifically, EXP utilizes a edit distance trick d_{edit} to search for the key with highest statistics, improving robustness in case of potential text alteration. For the mark module, EXP takes the private key as seed to sample a standard Gumbel vector $\mathbf{g}_{\mathbf{k}_i}$. Gumbel-max sampling is then conduct to sample a token t^*. Finally, the degenerate distribution $\delta(t^*)$ is returned as modified distribution.
- ITS (Kuditipudi et al., 2023). ITS utilizes the same key module as EXP. For the mark module, ITS takes the private key as seed to sample a standard uniform variable $\mathrm{u}_{\mathbf{k}_i}$ and a random permutation $\pi_{\mathbf{k}_i}$ for a permutational inverse transform sampling. The degenerate distribution of the sampled token is then returned as the modified distribution.

3.2 Behind Trade-offs: Conflicts within Key Module

In this section, we examine how the key module affects imperceptibility, efficacy and robustness. As stated in Section 2, the statistical difference between the null and alternative hypotheses is crucial for ensuring efficacy and robustness, i.e., whether the candidate token is sampled from a modified distribution. Regardless of the choice of statistic $S(\hat{\mathbf{k}}_i, \hat{\mathbf{y}})$, prior knowledge of the modified distribution is essential. Since the modification process is stochastic and only dependent on the private key, successful detection requires that the restored key $\hat{\mathbf{k}}_i$ matches the true key used during generation.

The key restoration process fundamentally involves searching through the potential key space Ξ. Both the ITS and EXP methods use a greedy search strategy, enumerating each potential key to identify the one exhibiting the highest statistic. While reliable in key restoration, it is markedly time-consuming. Moreover, the per-token statistic is now $\tilde{s} = \max s$, potentially diminishing the statistical difference. To mitigate the issue, ITS and EXP limit the possible key space size. To the extreme, Unigram directly fixes the private key. Alternatively, methods like KGW, Delta, and Gamma take the context window through a hash function for key restoration, reducing time complexity compared to exhaustive searches. But there are risks of incorrect key restoration if context is altered by attacks, diminishing their robustness.

As for imperceptibility under the key-centered scheme, Equation 2 can be rewritten as:

$$\prod_{i,n} P_M(\mathbf{y}_i^n|\mathbf{x}^n,\mathbf{y}_{<i}^n) = \mathbb{E}_{\mathbf{k}^1,\dots,\mathbf{k}^N}[\prod_{i,n} F(\mathbf{k}_i^n, P_i^n)(\mathbf{y}_i^n)]$$

We then propose two requirements:

Proposition 3.1. *A watermark is imperceptible if (1) Independent condition: the sampled private key vectors for each generated output are mutually independent, i.e.* $\mathbf{k}^1, ..., \mathbf{k}^N \overset{i.i.d}{\sim} \mathcal{U}(\mathbb{R}^L)$[5]*; (2) Unbiased condition: the modification function F satisfies* $P_M(\cdot|\mathbf{x}^n, \mathbf{y}_{<i}^n) = \mathbb{E}_{\mathbf{k}_i \sim \mathcal{U}(\mathbb{R})}[F(\mathbf{k}_i, P_i)]$.

The proposition describes separate requirements for key and mark modules. The detailed proof is presented in Appendix C.1. While many mark modules meet the unbiased condition (Kuditipudi et al., 2023; Hu et al., 2023), ensuring independent condition is challenging for key modules. The independence of private keys over the whole space $\mathbb{R}$ in successive generations indicates the search space Ξ of key restoration grows linearly with the number of generations, as all previously used keys must be considered.

The conflict within key modules now becomes apparent. Imperceptibility requires a large possible key space Ξ, which complicates the key restoration process during detection, thereby hindering both efficacy and robustness. This trade-off within the key module is a critical factor underlying the broader trade-off among the three properties.

3.3 WaterPool: Semantics-based Key Module

The previous section highlights a conflict within the key module, which contributes to the trade-off issue in watermarking. A natural question arises: is it possible to break the conflict for mitigating the trade-off? To ensure imperceptibility, it is essential to maintain a sufficiently large key space, characterized by an i.i.d. sampling strategy. Thus, an efficient and precise search strategy for the private key is needed for efficacy. The context-hash strategy provides insights by leveraging contextual information, yet hash functions are fragile to attacks. We aim to find a robust signal within the candidate's context that withstands semantic-preserved attacks.

[5] L is the maximum output length of LLM.

To this end, we propose **WaterPool**, a simple but effective key module empowered by semantic searching. Specifically, for each generation, WaterPool independently samples a private key vector $\mathbf{k}^n \sim \mathcal{U}(\mathbb{R}^L)$ to meet the independent condition in Proposition 3.1. Each private key is then used for the mark module to modify the next token distribution. We maintain a vector database $[(Enc(\mathbf{y}^1), \mathbf{k}^1), ..., (Enc(\mathbf{y}^N), \mathbf{k}^N)]$ to store the semantic embedding $Enc(\mathbf{y}^n)$ of each output as keys and the corresponding private key vector $\mathbf{k}^n$ as values[6]. For each candidate text $\hat{\mathbf{y}}$ during detection, regardless of whether it is watermarked, the most plausible private key vector $\hat{\mathbf{k}}$ is retrieved based on semantic similarity:

$$\hat{\mathbf{k}}{=}\mathbf{k}^{n^*}, \text{ where } n^*{=}\underset{n}{\mathrm{argmax}}\,\mathrm{sim}(Enc(\mathbf{y}^n),Enc(\hat{\mathbf{y}}))$$

The restored key vector is then provided for the mark module to calculate the statistics. Given that the most similar text to the candidate is its own even under attacks, this method ensures accurate key restoration if the candidate is watermarked.

As a key module, WaterPool is able to integrate with diverse mark modules. In this study, we integrate WaterPool with mark modules of three prominent watermarking techniques including EXP, ITS and KGW. These improved watermarks are referred to as EXP-Pool, ITS-Pool and KGW-Pool. Pseudo codes are presented in Appendix B.

Building on Proposition 3.1, the imperceptibility of WaterPool is readily established with EXP and ITS satisfying the unbiased condition [7].

Proposition 3.2. *Both ITS-Pool and EXP-Pool are imperceptible.*

Furthermore, the efficacy of WaterPool can also be assured based on the efficacy of combined mark module. This can be formalized as,

Proposition 3.3. *The statistical difference in WaterPool is bounded from below, as expressed by:*

$$\mathbb{E}[S(\hat{\mathbf{k}}_i, \mathbf{y}_i)|H_1]{-}\mathbb{E}[S(\hat{\mathbf{k}}_i, \mathbf{y}_i)|H_0] \geq p_{recall}{\cdot}\phi(\mathbf{p}^i)$$

,where $\phi(\boldsymbol{p})$ is watermarking potentials of the corresponding mark module depending on the probability vector $\mathbf{p}^i$ for the i-th token distribution $P_M(\cdot|\mathbf{x}, \mathbf{y}_{<i})$.

This proposition indicates that WaterPool can effectively leverage the power of mark modules (i.e. the lower bound of its statistical difference with golden private key restoration), slightly modulated by the recall performance p_{recall} of WaterPool's retriever. We will empirically demonstrate that the recall performance is near-optimal even in large scale database employing relatively weak retrievers in the following experiments. Proofs of propositions above are presented in Appendix C.2 to C.6.

3.4 Difference from Retrieval Watermark

Krishna et al. (2023) proposed a retrieval watermark to distinguish watermarked texts with semantic retrieval. Different from the key-centered scheme described in Section 3.1, they directly store every output o generated by the specific LLM to be "watermarked" in a vector database D. During detection, they search over the database for similar items, and utilize $\max_{o \in D} \mathrm{sim}(o, o^{\text{candidate}})$ as confidence of the candidate being watermarked.

WaterPool fundamentally differs from retrieval watermark, though they both leverage retrieval techniques. WaterPool's efficacy and robustness rely primarily on the statistical difference guaranteed by the mark module. In specific, $\mathbb{E}[S(\mathbf{k}_i, \mathbf{y}_i)]$ is designed to be high if $\mathbf{y}_i$ is sampled from $\mathbf{k}_i$-induced modified distribution and low if $\mathbf{k}_i$ and $\mathbf{y}_i$ are independent. Therefore, the retriever of WaterPool only needs to retrieve the correct private key for watermarked candidates, without concern for retrieval results of non-watermarked candidates, since all keys stored in the database are independent of all non-watermarked texts.

In contrast, retrieval watermark directly uses similarity as score. For efficacy, it should ensure high similarity scores for watermarked texts and low similarity scores for non-watermarked ones. However, due to the dense semantic space of human language, texts stored in the database often have similar non-watermarked neighbors (*semantic collisions* in Krishna et al. (2023)), and hence reducing efficacy. This issue will become more severe as the number of non-watermarked samples or the size of the vector database increases. We conduct an experiment to empirically demonstrate this by utilizing responses of other models to the same prompts as non-watermarked samples in Appendix E.2. In this scenario, the performance of retrieval watermark is over 40% worse than other watermarking methods.

[6] Enc can be any semantic embedding models, e.g. BERT.

[7] The mark module of KGW does not satisfy the unbiased condition. Therefore, WaterPool can only enhance its imperceptibility performance, but not achieve optimal imperceptibility.

	Glob-distinct2		Glob-distinct3		Group-distinct2		Group-distinct3		Perplexity	
	value↑	Δ↑	value↑	Δ↑	value↑	Δ↑	value↑	Δ↑	value↓	Δ↓
Open-Ended Text Generation										
Non-watermark	38.8$_{\pm 0.0}$	0.0$_{\pm 0.0}$	76.2$_{\pm 0.0}$	0.0$_{\pm 0.0}$	86.2$_{\pm 0.0}$	0.0$_{\pm 0.0}$	96.2$_{\pm 0.0}$	0.0$_{\pm 0.0}$	7.8$_{\pm 0.0}$	0.0$_{\pm 0.0}$
Gamma	38.8$_{\pm 0.0}$	0.0$_{\pm 0.0}$	76.2$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	86.2$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	96.2$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	7.8$_{\pm 0.0}$	0.0$_{\pm 0.0}$
Delta	38.8$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	76.2$_{\pm 0.0}$	-0.0$_{\pm 0.1}$	86.2$_{\pm 0.0}$	-0.0$_{\pm 0.1}$	96.2$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	7.8$_{\pm 0.0}$	0.0$_{\pm 0.0}$
Unigram	33.4$_{\pm 1.9}$	-5.3$_{\pm 1.9}$	69.9$_{\pm 2.8}$	-6.3$_{\pm 2.7}$	82.6$_{\pm 2.4}$	-3.6$_{\pm 2.4}$	95.1$_{\pm 0.6}$	-1.1$_{\pm 0.6}$	9.9$_{\pm 0.6}$	2.2$_{\pm 0.6}$
KGW	36.7$_{\pm 0.1}$	-2.0$_{\pm 0.2}$	73.6$_{\pm 0.1}$	-2.7$_{\pm 0.1}$	85.5$_{\pm 0.1}$	-0.7$_{\pm 0.1}$	95.9$_{\pm 0.0}$	-0.3$_{\pm 0.1}$	9.6$_{\pm 0.0}$	1.8$_{\pm 0.0}$
KGW-Pool	**40.5**$_{\pm 0.2}$	**1.8**$_{\pm 0.2}$	**78.7**$_{\pm 0.2}$	**2.4**$_{\pm 0.2}$	**87.3**$_{\pm 0.1}$	**1.1**$_{\pm 0.2}$	**96.7**$_{\pm 0.0}$	**0.5**$_{\pm 0.1}$	9.9$_{\pm 0.0}$	2.1$_{\pm 0.0}$
EXP	30.2$_{\pm 0.0}$	-8.5$_{\pm 0.0}$	59.2$_{\pm 0.0}$	-17.0$_{\pm 0.0}$	73.4$_{\pm 0.1}$	-12.8$_{\pm 0.0}$	82.2$_{\pm 0.1}$	-14.0$_{\pm 0.0}$	7.8$_{\pm 0.0}$	0.0$_{\pm 0.0}$
EXP-Pool	38.7$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	76.2$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	86.2$_{\pm 0.0}$	-0.0$_{\pm 0.1}$	96.2$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	7.8$_{\pm 0.0}$	0.0$_{\pm 0.0}$
ITS	34.4$_{\pm 0.7}$	-4.4$_{\pm 0.7}$	66.4$_{\pm 1.5}$	-9.8$_{\pm 1.5}$	75.2$_{\pm 1.8}$	-11.0$_{\pm 1.8}$	83.7$_{\pm 2.1}$	-12.6$_{\pm 2.1}$	**7.5**$_{\pm 0.0}$	**-0.3**$_{\pm 0.0}$
ITS-Pool	38.8$_{\pm 0.0}$	0.0$_{\pm 0.0}$	76.2$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	86.2$_{\pm 0.0}$	0.0$_{\pm 0.0}$	96.2$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	7.8$_{\pm 0.0}$	0.0$_{\pm 0.0}$
Long-Form Question Answering										
Non-watermark	31.5$_{\pm 0.0}$	0.0$_{\pm 0.0}$	70.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	86.7$_{\pm 0.0}$	0.0$_{\pm 0.0}$	97.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	9.5$_{\pm 0.0}$	0.0$_{\pm 0.0}$
Gamma	31.5$_{\pm 0.0}$	0.0$_{\pm 0.0}$	70.0$_{\pm 0.0}$	0.0$_{\pm 0.1}$	86.7$_{\pm 0.0}$	0.0$_{\pm 0.0}$	97.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	9.5$_{\pm 0.0}$	0.0$_{\pm 0.0}$
Delta	31.5$_{\pm 0.0}$	0.0$_{\pm 0.0}$	70.0$_{\pm 0.0}$	-0.0$_{\pm 0.1}$	86.8$_{\pm 0.0}$	0.0$_{\pm 0.0}$	97.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	9.5$_{\pm 0.0}$	0.0$_{\pm 0.0}$
Unigram	26.4$_{\pm 2.0}$	-5.1$_{\pm 2.0}$	62.0$_{\pm 2.9}$	-7.9$_{\pm 2.9}$	81.2$_{\pm 2.5}$	-5.5$_{\pm 2.5}$	94.8$_{\pm 0.6}$	-2.2$_{\pm 0.6}$	10.9$_{\pm 1.4}$	1.4$_{\pm 1.4}$
KGW	29.5$_{\pm 0.2}$	-2.0$_{\pm 0.2}$	66.2$_{\pm 0.3}$	-3.8$_{\pm 0.2}$	85.4$_{\pm 0.1}$	-1.3$_{\pm 0.1}$	96.4$_{\pm 0.0}$	-0.7$_{\pm 0.1}$	11.6$_{\pm 0.1}$	2.1$_{\pm 0.1}$
KGW-Pool	**32.9**$_{\pm 0.2}$	**1.4**$_{\pm 0.2}$	**71.6**$_{\pm 0.3}$	**1.6**$_{\pm 0.3}$	83.8$_{\pm 0.3}$	-2.9$_{\pm 0.3}$	94.3$_{\pm 0.2}$	-2.8$_{\pm 0.2}$	10.9$_{\pm 0.1}$	1.4$_{\pm 0.1}$
EXP	22.6$_{\pm 0.3}$	-8.9$_{\pm 0.3}$	50.0$_{\pm 0.7}$	-20.0$_{\pm 0.7}$	75.3$_{\pm 1.1}$	-11.4$_{\pm 1.1}$	85.1$_{\pm 1.4}$	-11.9$_{\pm 1.4}$	9.6$_{\pm 0.0}$	0.1$_{\pm 0.0}$
EXP-Pool	31.5$_{\pm 0.0}$	0.0$_{\pm 0.0}$	70.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	**86.8**$_{\pm 0.0}$	**0.1**$_{\pm 0.0}$	**97.1**$_{\pm 0.0}$	**0.0**$_{\pm 0.0}$	9.5$_{\pm 0.0}$	0.0$_{\pm 0.0}$
ITS	27.8$_{\pm 0.6}$	-3.7$_{\pm 0.6}$	60.7$_{\pm 1.4}$	-9.3$_{\pm 1.4}$	76.1$_{\pm 1.9}$	-10.6$_{\pm 1.9}$	84.7$_{\pm 2.3}$	-12.3$_{\pm 2.3}$	**9.1**$_{\pm 0.0}$	**-0.4**$_{\pm 0.0}$
ITS-Pool	31.5$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	69.9$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	86.8$_{\pm 0.0}$	0.1$_{\pm 0.0}$	97.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	9.8$_{\pm 0.0}$	0.3$_{\pm 0.0}$

Table 1: Imperceptibility of different watermarking methods on OPT-1.3B. Δ is the difference between watermarked and non-watermarked texts. The best and second-best results before rounding are highlighted in **bold** and underline.

4 Experiments

Datasets. Following Kirchenbauer et al. (2023a,b), we include two common used datasets, the C4 dataset and "Explain Like I'm Five" (ELI5) (Fan et al., 2019) for open-ended text generation and long-form question answering, respectively. We randomly select about 3000 texts from both datasets as prompts for two LLMs, OPT-1.3b and OPT-6.7b, following Krishna et al. (2023).

Metrics. We generate 20 watermarked outputs for each prompt, while considering outputs of the original LLM as non-watermarked. Subsequently, 120,000 samples are used to evaluate each method. For both efficacy and robustness, we report true positive rate at 1% false positive rate, denoted as *TPR@FPR=1%*. We also include *ROC-AUC* in Appendix E. To evaluate the robustness of watermarking techniques, we include three kinds of attacks, namely Lexical-Attack, Dipper-Attack and Translation-Attack. Lexical-attack randomly add/delete/replace 10% tokens of texts. Dipper is a paraphrasing model (Krishna et al., 2023). Translation-attack represents roundtrip-translation, which is a paraphrasing pipeline. For the evaluation of imperceptibility, we split the criteria into two aspects: (1) the distribution bias within each output (2) the independence among different outputs. The former can be evaluated with *Perplexity* while the latter can be roughly evaluated with n-gram distinction (Kirchenbauer et al., 2023b). Specifically, we consider the distinction across all outputs (*Glob-distinct-N*) and within outputs in response to one single prompt (*Group-distinct-N*).

Baselines and implementation details. We include several typical methods as baselines. In addition to EXP, ITS and KGW, we also include Gamma (Hu et al., 2023), Delta (Hu et al., 2023) and Unigram (Zhao et al., 2023). All baselines are reproduced based on source codes provided by original paper. We use a 128 dimension sentence embedding model (Nussbaum et al., 2024) as $Enc(\cdot)$ in WaterPool. For implementation of mark modules in different WaterPool (i.e. KGW-Pool, ITS-Pool, EXP-Pool), we use identical hyperparameter settings as the corresponding baselines. Other details are presented in Appendix D.

4.1 Main Results

We here only present and discuss results of OPT-1.3b in this section. Results of OPT-6.7b and more analysis experiments are presented in Appendix E. Notably, results across different LLMs on different tasks exhibit consistent trends and patterns.

Most baselines have achieved high efficacy (over 90% TPR@FPR=1%), but they all make trade-offs between imperceptibility and robustness. Consistent with our theoretical analysis in Section 3.2, Unigram, EXP, and ITS limit the key space Ξ, thereby reducing the independence of keys across different generations. This is reflected in their weaker performance on the Distinct-N metric. Specifically,

	w/o Attack		Lexical-Attack		Dipper-Attack		Translation-Attack	
	value↑	Δ	value↑	Δ	value↑	Δ	value↑	Δ
Open-Ended Text Generation								
Gamma	$96.94_{\pm0.05}$	-	$17.91_{\pm0.28}$	-	$2.26_{\pm0.04}$	-	$3.25_{\pm0.10}$	-
Delta	$75.37_{\pm0.34}$	-	$8.58_{\pm0.27}$	-	$2.07_{\pm0.08}$	-	$2.91_{\pm0.11}$	-
Unigram	$93.98_{\pm1.32}$	-	$89.69_{\pm3.67}$	-	$19.99_{\pm7.44}$	-	$35.35_{\pm8.74}$	-
KGW	$\mathbf{98.43}_{\pm0.08}$	-	$88.88_{\pm0.13}$	-	$15.05_{\pm0.32}$	-	$29.53_{\pm0.23}$	-
KGW-Pool	$98.29_{\pm0.01}$	$-0.15_{\pm0.08}$	$\underline{95.29_{\pm1.04}}$	$6.41_{\pm1.17}$	$\underline{24.62_{\pm2.01}}$	$9.57_{\pm2.32}$	$\underline{42.26_{\pm1.70}}$	$12.73_{\pm1.73}$
EXP	$97.19_{\pm0.08}$	-	$93.48_{\pm0.09}$	-	$18.32_{\pm0.36}$	-	$31.14_{\pm0.33}$	-
EXP-Pool	$\underline{98.43_{\pm0.01}}$	$1.24_{\pm0.09}$	$\mathbf{96.67}_{\pm0.07}$	$3.19_{\pm0.15}$	$\mathbf{26.17}_{\pm0.86}$	$7.85_{\pm0.62}$	$\mathbf{51.41}_{\pm0.42}$	$20.27_{\pm0.64}$
ITS	$73.43_{\pm0.10}$	-	$26.23_{\pm0.19}$	-	$2.16_{\pm0.08}$	-	$3.56_{\pm0.08}$	-
ITS-Pool	$92.56_{\pm0.14}$	$19.12_{\pm0.11}$	$68.50_{\pm0.46}$	$42.27_{\pm0.40}$	$4.05_{\pm0.15}$	$1.89_{\pm0.11}$	$10.83_{\pm0.31}$	$7.27_{\pm0.39}$
Long-Form Question Answering								
Gamma	$98.68_{\pm0.05}$	-	$21.20_{\pm0.36}$	-	$2.31_{\pm0.05}$	-	$5.18_{\pm0.21}$	-
Delta	$90.19_{\pm0.13}$	-	$12.17_{\pm0.10}$	-	$2.21_{\pm0.08}$	-	$4.96_{\pm0.02}$	-
Unigram	$96.93_{\pm1.99}$	-	$92.47_{\pm3.77}$	-	$26.38_{\pm5.88}$	-	$43.17_{\pm6.98}$	-
KGW	$\underline{99.51_{\pm0.01}}$	-	$94.12_{\pm0.06}$	-	$19.21_{\pm0.17}$	-	$46.62_{\pm0.43}$	-
KGW-Pool	$99.51_{\pm0.00}$	$-0.01_{\pm0.01}$	$\underline{97.97_{\pm0.04}}$	$3.85_{\pm0.08}$	$\underline{29.92_{\pm1.04}}$	$10.71_{\pm1.20}$	$50.14_{\pm0.24}$	$3.52_{\pm0.37}$
EXP	$99.17_{\pm0.06}$	-	$97.56_{\pm0.08}$	-	$27.92_{\pm0.56}$	-	$\underline{54.99_{\pm0.37}}$	-
EXP-Pool	$\mathbf{99.56}_{\pm0.04}$	$0.40_{\pm0.08}$	$\mathbf{98.81}_{\pm0.07}$	$1.25_{\pm0.05}$	$\mathbf{36.24}_{\pm0.82}$	$8.32_{\pm1.37}$	$\mathbf{72.61}_{\pm0.29}$	$17.62_{\pm0.56}$
ITS	$86.40_{\pm0.51}$	-	$38.23_{\pm0.46}$	-	$3.02_{\pm0.19}$	-	$8.19_{\pm0.17}$	-
ITS-Pool	$97.56_{\pm0.05}$	$11.16_{\pm0.46}$	$81.73_{\pm0.34}$	$43.51_{\pm0.21}$	$6.25_{\pm0.15}$	$3.23_{\pm0.05}$	$24.26_{\pm0.43}$	$16.07_{\pm0.27}$

Table 2: Efficacy and robustness of different watermarking methods on OPT-1.3B evaluated with TPR@FPR=1%. Δ is the performance boost brought by WaterPool. The best and second-best results are shown in **bold** and underline.

Unigram suffers from degradation in both Distinct-N and perplexity. While EXP and ITS perform well in perplexity, they exhibit the worst Distinct-N results with more than 10% degradation. On the contrary, Gamma and Delta achieve optimal imperceptibility but at the cost of both efficacy and robustness. During detection, the alternative hypothesis distribution required by their likelihood ratio test is often inaccurate due to missing original prompts. Thus, their TPR@FPR=1% is significantly weaker compared to other methods.

WaterPool effectively mitigates these trade-offs. It markedly improves the original watermarking techniques across all tasks and all three aspects. It elevates the imperceptibility of original watermarks to near-optimal levels, as evidenced by the minimal difference from non-watermarked texts. Moreover, it consistently enhances the robustness of the original watermarks, as shown by substantial improvements in the TPR@FPR=1% metric (e.g. KGW-Pool outperforms KGW by 12.73%, EXP-Pool outperforms EXP by 20.27%, ITS-Pool outperforms ITS by 7.27%).

4.2 Real-world Challenges for WaterPool

As stated in Section 3.4, WaterPool only requires to retrieve golden private key if it exists. This assertion holds intuitively, as a watermarked text, even under attacks, should remain semantically closer to the original watermarked text stored in the database than other texts. Otherwise, it should not be considered a modified version of the original watermarked text. To empirically demonstrate this, we conduct various experiments to demonstrate the stability of WaterPool under two real-world challenges.

Performance with Diverse Negative Samples. We test the performance of WaterPool across various distributions of non-watermarked text, including human-written outputs (*Human*, 3K samples) and outputs from other non-watermarked models[8] (*Other Models*, 1.8M samples) with respect to the same prompts. Results shown in Table 3 indicate that all WaterPool methods exhibit stable performance regardless of the number and types of negative samples, which aligns with our theoretical analysis. This stability also highlights the advantages of WaterPool over retrieval watermarking, further substantiating the claims in Section 3.4.

Scalability with Large Vector Databases. Another real-world challenge for WaterPool is the increasing scale of possible key space Ξ with increasing calls to watermarked LLMs. This raises the question of whether the key module can accurately retrieve the correct key from a large-scale database. To this end, we conduct experiments by scaling up the vector database size to simulate this scenario. Specifically, we augment the database with noisy entries by incorporating 50-token fragments sampled from the C4 dataset, constructing a noisy

[8] Models include Gemma-2b, Gemma-7b, Llama2-7b, Llama2-13b, Vicuna-7b and Vicuna-13b.

	Original	Human	Other Models
w/o Attack	98.43	98.35	98.45
Lexical-Attack	96.67	96.50	96.72
Dipper-Attack	26.17	24.49	25.65
Translation-Attack	51.41	50.23	51.60

Table 3: TPR@FPR=1% of EXP-Pool with different non-watermarked texts listed in the first row. WaterPool remains stable across different non-watermarked text sources. Full results are presented in Appendix E.3.

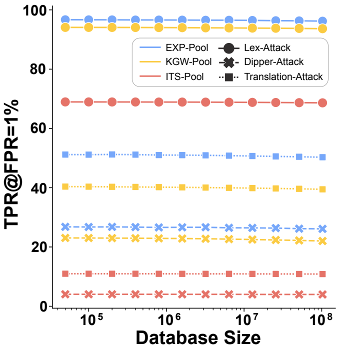

Figure 3: TPR@FPR=1% of WaterPool with different database size. WaterPool maintains stable performance as the scale increases to 100M items exponentially.

database of more than 100 million items. During detection, if WaterPool retrieves a noisy item from the fake database, representing a failed key reconstruction, a random key is used for the per-token statistic calculation, thereby affecting its efficacy and robustness. This setting is particularly challenging for open-ended text generation task since the noisy database shares a similar distribution with the watermarked texts stored in real database, both sampled from C4 dataset. Results on open-ended text generation and long-form question answering are presented in Figure 3. The results demonstrate that WaterPool maintains robust performance even as the database size increases exponentially, indicating its feasibility in real-world applications.

4.3 Analysis of Space and Time

The price for breaking the trade-off issues with WaterPool lies in its space usage. The space complexity of WaterPool is $O(N)$, growing linearly with the number of generations. However, this is still practical in real-world scenarios. In our experiments, we use sentence embeddings in form of 128-dimensional bfloat16 arrays and a private key of one int32 number[9]. Given ChatGPT's monthly visits are about 2B per month[10], it takes 260 bytes of space per item, resulting in about 520 GB of storage per month, which is certainly manageable nowadays. Regarding time complexity, WaterPool requires less than 0.001 sec per item to retrieve from a database of 100M items with 10 RTX3090 GPUs, which is sufficient since watermark detection is not a time-intensive application.

5 Related Work

Watermarking is a specific form of steganography. Steganography requires that, without the knowledge about private keys, distributions of original texts and texts with steganography must be indistinguishable (Simmons, 1984; Katzenbeisser and Petitcolas, 1999; Hopper et al., 2009; Dedić et al., 2008; Fang et al., 2017), which leads to imperceptibility objective of watermarking in Section 2.

LLM watermarking has recently gained many attention (Christ et al., 2023; Kuditipudi et al., 2023; Hu et al., 2023; Pri, 2023; Zhao et al., 2024; Christ and Gunn, 2024; Fairoze et al., 2023; He et al., 2024), which can be seamlessly integrated into all LLMs without further training. KGW (Kirchenbauer et al., 2023a) is the pioneering work, using context windows as private keys to increase probabilities of a specific partition of vocabulary. Building on KGW, follow-ups made many improvements, such as proposing different hash functions (Kirchenbauer et al., 2023b; Hou et al., 2023; Liu et al., 2023a; Ren et al., 2023), heuristic partition strategies (Li et al., 2023; Chen et al., 2023), embedding multi-bit messages (Wang et al., 2023; Qu et al., 2024), and robust hypothesis testing techniques (Fernandez et al., 2023). Despite these advancements, the trade-off among imperceptibility, efficacy, and robustness has been widely recognized and remains unresolved. A concurrent work (Giboulot and Teddy, 2024) also tries to break the trade-off by resampling until observing significant watermarking signals. Although this approach maintains imperceptibility in a single turn, it significantly alters the N-shot output distribution.

6 Conclusion

In this paper, we focus on the trade-off challenges among imperceptibility, efficacy and robustness in LLM watermarking. Through a key-centered scheme, we have identified that the trade-offs arise

[9] In practice, we can only sample a number as seed to initialize a pseudo-random number generator, thereby generating the whole private key vector $\mathbf{k} \in \mathbb{R}^L$ for all tokens.

[10] https://explodingtopics.com/blog/chatgpt-users

from the conflict between the scale of the key sampling space during generation and the complexity of key restoration during detection. This insight motivates the design of WaterPool, a key module utilizing semantic search to alleviate this conflict. WaterPool integrates seamlessly with most existing watermarking methods, significantly enhancing their performance across all three dimensions. We hope this work offers valuable insights to future research for better solutions to this challenge.

Acknowledgments

This work was supported by Beijing Science and Technology Program (Z231100007423011) 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.

Limitations

While WaterPool is both simple and easy to deploy, it does not entirely resolve the trade-offs between imperceptibility, efficacy, and robustness. The mark module is also a pivotal component in achieving these goals. We have defined the unbiased condition and statistical differences to outline the requirements for mark modules. However, the specific design and optimization of mark modules are not discussed in this paper. Future research could explore advanced mark module designs, advancing towards fully resolving the trade-off challenges in LLM watermarking.

References

2023. A Private Watermark for Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Liang Chen, Yatao Bian, Yang Deng, Shuaiyi Li, Bingzhe Wu, Peilin Zhao, and Kam-fai Wong. 2023. X-Mark: Towards Lossless Watermarking Through Lexical Redundancy. *Preprint*, arXiv:2311.09832.

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

Miranda Christ and Sam Gunn. 2024. Pseudorandom Error-Correcting Codes. *Preprint*, arXiv:2402.09370.

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable Watermarks for Language Models. *Preprint*, arXiv:2306.09194.

Nenad Dedić, Gene Itkis, Leonid Reyzin, and Scott Russell. 2008. Upper and lower bounds on black-box steganography. *Journal of Cryptology*, 22(3):365–394.

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. 2023. Publicly Detectable Watermarking for Language Models. *Preprint*, arXiv:2310.18491.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Tina Fang, Martin Jaggi, and Katerina Argyraki. 2017. Generating steganographic text with LSTMs. In *Proceedings of ACL 2017, Student Research Workshop*, pages 100–106, Vancouver, Canada. Association for Computational Linguistics.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. 2023. Three Bricks to Consolidate Watermarks for Large Language Models. *Preprint*, arXiv:2308.00113.

Eva Giboulot and Furon Teddy. 2024. WaterMax: Breaking the LLM watermark detectability-robustness-quality trade-off. *Preprint*, arXiv:2403.04808.

Zhiwei He, Binglin Zhou, Hongkun Hao, Aiwei Liu, Xing Wang, Zhaopeng Tu, Zhuosheng Zhang, and Rui Wang. 2024. Can watermarks survive translation? on the cross-lingual consistency of text watermark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4115–4129, Bangkok, Thailand. Association for Computational Linguistics.

Nicholas Hopper, Luis von Ahn, and John Langford. 2009. Provably secure steganography. *IEEE Transactions on Computers*, 58(5):662–676.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. SemStamp: A Semantic Watermark with Paraphrastic Robustness for Text Generation. *Preprint*, arXiv:2310.03991.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2023. Unbiased Watermark for Large Language Models. *Preprint*, arXiv:2310.10669.

Stephan Katzenbeisser and Fabien Petitcolas. 1999. *Information Hiding Techniques for Steganography and Digital Watermaking*, volume 28.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A Watermark for Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17061–17084. PMLR.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2023b. On the Reliability of Watermarks for Large Language Models. *Preprint*, arXiv:2306.04634.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. *Preprint*, arXiv:2303.13408.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust Distortion-free Watermarks for Language Models. *Preprint*, arXiv:2307.15593.

Yuhang Li, Yihan Wang, Zhouxing Shi, and Cho-Jui Hsieh. 2023. Improving the Generation Quality of Watermarked Large Language Models via Word Importance Scoring. *Preprint*, arXiv:2311.09668.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023a. A Semantic Invariant Robust Watermark for Large Language Models. *Preprint*, arXiv:2310.06356.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. 2023b. A Survey of Text Watermarking in the Era of Large Language Models. *Preprint*, arXiv:2312.07913.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder. *Preprint*, arXiv:2402.01613.

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. 2023. Mark My Words: Analyzing and Evaluating Language Model Watermarks. *Preprint*, arXiv:2312.00273.

Wenjie Qu, Dong Yin, Zixin He, Wei Zou, Tianyang Tao, Jinyuan Jia, and Jiaheng Zhang. 2024. Provably Robust Multi-bit Watermarking for AI-generated Text via Error Correction Code. *Preprint*, arXiv:2401.16820.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2023. A Robust Semantics-based Watermark for Large Language Model against Paraphrasing. *Preprint*, arXiv:2311.08721.

Gustavus J. Simmons. 1984. *The Prisoners' Problem and the Subliminal Channel*, pages 51–67. Springer US, Boston, MA.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, and Surya Bhupatiraju et al. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, and et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Towards Codable Text Watermarking for Large Language Models. *Preprint*, arXiv:2307.15992.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable Robust Watermarking for AI-Generated Text. https://arxiv.org/abs/2306.17439v2.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. 2024. Permute-and-Flip: An optimally robust and watermarkable decoder for LLMs. *Preprint*, arXiv:2402.05864.

References

2023. A Private Watermark for Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Liang Chen, Yatao Bian, Yang Deng, Shuaiyi Li, Bingzhe Wu, Peilin Zhao, and Kam-fai Wong. 2023. X-Mark: Towards Lossless Watermarking Through Lexical Redundancy. *Preprint*, arXiv:2311.09832.

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

Miranda Christ and Sam Gunn. 2024. Pseudorandom Error-Correcting Codes. *Preprint*, arXiv:2402.09370.

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable Watermarks for Language Models. *Preprint*, arXiv:2306.09194.

Nenad Dedić, Gene Itkis, Leonid Reyzin, and Scott Russell. 2008. Upper and lower bounds on black-box steganography. *Journal of Cryptology*, 22(3):365–394.

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. 2023. Publicly Detectable Watermarking for Language Models. *Preprint*, arXiv:2310.18491.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Tina Fang, Martin Jaggi, and Katerina Argyraki. 2017. Generating steganographic text with LSTMs. In *Proceedings of ACL 2017, Student Research Workshop*, pages 100–106, Vancouver, Canada. Association for Computational Linguistics.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. 2023. Three Bricks to Consolidate Watermarks for Large Language Models. *Preprint*, arXiv:2308.00113.

Eva Giboulot and Furon Teddy. 2024. WaterMax: Breaking the LLM watermark detectability-robustness-quality trade-off. *Preprint*, arXiv:2403.04808.

Zhiwei He, Binglin Zhou, Hongkun Hao, Aiwei Liu, Xing Wang, Zhaopeng Tu, Zhuosheng Zhang, and Rui Wang. 2024. Can watermarks survive translation? on the cross-lingual consistency of text watermark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4115–4129, Bangkok, Thailand. Association for Computational Linguistics.

Nicholas Hopper, Luis von Ahn, and John Langford. 2009. Provably secure steganography. *IEEE Transactions on Computers*, 58(5):662–676.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. SemStamp: A Semantic Watermark with Paraphrastic Robustness for Text Generation. *Preprint*, arXiv:2310.03991.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2023. Unbiased Watermark for Large Language Models. *Preprint*, arXiv:2310.10669.

Stephan Katzenbeisser and Fabien Petitcolas. 1999. *Information Hiding Techniques for Steganography and Digital Watermaking*, volume 28.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A Watermark for Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17061–17084. PMLR.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2023b. On the Reliability of Watermarks for Large Language Models. *Preprint*, arXiv:2306.04634.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. *Preprint*, arXiv:2303.13408.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust Distortion-free Watermarks for Language Models. *Preprint*, arXiv:2307.15593.

Yuhang Li, Yihan Wang, Zhouxing Shi, and Cho-Jui Hsieh. 2023. Improving the Generation Quality of Watermarked Large Language Models via Word Importance Scoring. *Preprint*, arXiv:2311.09668.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023a. A Semantic Invariant Robust Watermark for Large Language Models. *Preprint*, arXiv:2310.06356.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. 2023b. A Survey of Text Watermarking in the Era of Large Language Models. *Preprint*, arXiv:2312.07913.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder. *Preprint*, arXiv:2402.01613.

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. 2023. Mark My Words: Analyzing and Evaluating Language Model Watermarks. *Preprint*, arXiv:2312.00273.

Wenjie Qu, Dong Yin, Zixin He, Wei Zou, Tianyang Tao, Jinyuan Jia, and Jiaheng Zhang. 2024. Provably Robust Multi-bit Watermarking for AI-generated Text via Error Correction Code. *Preprint*, arXiv:2401.16820.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2023. A Robust Semantics-based Watermark for Large Language Model against Paraphrasing. *Preprint*, arXiv:2311.08721.

Gustavus J. Simmons. 1984. *The Prisoners' Problem and the Subliminal Channel*, pages 51–67. Springer US, Boston, MA.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, and Surya Bhupatiraju et al. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, and et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Towards Codable Text Watermarking for Large Language Models. *Preprint*, arXiv:2307.15992.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable Robust Watermarking for AI-Generated Text. https://arxiv.org/abs/2306.17439v2.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. 2024. Permute-and-Flip: An optimally robust and watermarkable decoder for LLMs. *Preprint*, arXiv:2402.05864.

Name	Reweight Function $F(\mathbf{k}_i, P_i)$	Statistic $S(\hat{\mathbf{k}}_i, \hat{\mathbf{y}}_i)$	Used in
logits-add	$\forall t \in \Sigma, \hat{L}_i(t) = L_i(t) + \tau \cdot \mathbf{1}_{t \in \mathcal{G}_{\mathbf{k}_i}}$ $F(\mathbf{k}_i, P_i) = \text{Softmax}(\hat{L}_i)$	$S(\hat{\mathbf{k}}_i, \hat{\mathbf{y}}_i) = \frac{\mathbf{1}_{\hat{\mathbf{y}}_i \in \mathcal{G}_{\hat{\mathbf{k}}_i}} - \gamma}{\sqrt{\text{len}(\mathbf{y})\gamma(1-\gamma)}}$	KGW(Kirchenbauer et al., 2023a), Unigram(Zhao et al., 2023)
inverse-sample	$\mathrm{u}_{\mathbf{k}_i} \sim \mathcal{U}([0,1])$ $\forall t \in \Sigma, P_i^{\text{perm}}(t) = P_i(\pi_{\mathbf{k}_i}^{-1}(t))$ $z^* = \min\{z \in \mathbb{Z} : \sum_{\text{ord}(t) \leq z} P_i^{\text{perm}}(t) > \mathrm{u}_{\mathbf{k}_i}\}$ $t^* = \pi_{\mathbf{k}_i}^{-1} \circ \text{ord}^{-1}(z^*)$ $F(\mathbf{k}_i, P_i) = \delta(t^*)$	$S(\hat{\mathbf{k}}_i, \hat{\mathbf{y}}_i) = (\mathrm{u}_{\hat{\mathbf{k}}_i} - \frac{1}{2}) \cdot (\frac{\text{ord}(\pi_{\hat{\mathbf{k}}_i}(\hat{\mathbf{y}}_i)) - 1}{\lvert\Sigma\rvert - 1} - \frac{1}{2})$	ITS(Kuditipudi et al., 2023)
gumbel-sample	$\mathbf{g}_{\mathbf{k}_i} \in \mathbb{R}^{\lvert\Sigma\rvert}, \mathbf{g}_{\mathbf{k}_i,j} \overset{\text{iid}}{\sim} \text{Gumbel(0,1)}$ $t^* = \arg\max_{t \in \Sigma} \log P_i(t) + \mathbf{g}_{\mathbf{k}_i,\text{ord}(t)}$ $F(\mathbf{k}_i, P_i) = \delta(t^*)$	$S(\hat{\mathbf{k}}_i, \hat{\mathbf{y}}_i) = -\exp(-\mathbf{g}_{\mathbf{k}_i,\text{ord}(\hat{\mathbf{y}}_i)})$	EXP(Kuditipudi et al., 2023)
prob-scale	$\forall t \in \Sigma, P_i^{\text{perm}}(t) = P_i(\pi_{\mathbf{k}_i}^{-1}(t))$ $\forall t \in \Sigma, C_i^{\text{perm}}(t) = \sum_{\text{ord}(t') < \text{ord}(t)} P_i^{\text{perm}}(t')$ $\forall t \in \Sigma, \hat{C}_i(t) = \min(2C_i^{\text{perm}}(t), 1)$ $\forall t \in \Sigma, \hat{P}_i^{\text{perm}}(t) = \hat{C}_i(t) - \hat{C}_i(\text{ord}^{-1}(\text{ord}(t) - 1))$ $\forall t \in \Sigma, \hat{P}_i(t) = \hat{P}_i^{\text{perm}}(\pi_{\mathbf{k}_i}(t))$ $F(\mathbf{k}_i, P_i) = \hat{P}_i$	$\forall t \in \Sigma, P_i^{\text{perm}}(t) = P_i(\pi_{\hat{\mathbf{k}}_i}^{-1}(t))$ $\forall t \in \Sigma, C_i^{\text{perm}}(t) = \sum_{\text{ord}(t') < \text{ord}(t)} P_i^{\text{perm}}(t')$ $\forall t \in \Sigma, \hat{C}_i(t) = \min(2C_i^{\text{perm}}(t), 1)$ $\forall t \in \Sigma, \hat{P}_i^{\text{perm}}(t) = \hat{C}_i(t) - \hat{C}_i(\text{ord}^{-1}(\text{ord}(t) - 1))$ $\forall t \in \Sigma, \hat{P}_i(t) = \hat{P}_i^{\text{perm}}(\pi_{\hat{\mathbf{k}}_i}(t))$ $S(\hat{\mathbf{k}}_i, \hat{\mathbf{y}}_i) = \log \hat{P}_i(\hat{\mathbf{y}}_i) - \log P_i(\hat{\mathbf{y}}_i)$	Gamma(Kuditipudi et al., 2023)

Table 4: Mark modules of typical watermarks. $P_i(\cdot)$, $C_i(\cdot)$ and $L_i(\cdot)$ are probability distribution function, cumulative distribution function and logit function of the i-th step generation. $ord : \Sigma \to \{1, ..., |\Sigma|\}$ is a function mapping each token to its order in the vocabulary. $\delta(t)$ represents a degeneration distribution taking only one value t. Subscript "$\mathbf{k}_i$" indicates a random variable seeded by $\mathbf{k}_i$. For example, $\mathcal{G}_{\mathbf{k}_i}$ is a random γ ratio partition of vocabulary, $\pi_{\mathbf{k}_i} : \Sigma \to \Sigma$ represents a random permutation over the vocabulary. Superscript "$perm$" indicates a distribution function after vocabulary permutation, such as P^{perm}. Other notations like γ, τ are fixed hyper-parameters.

Name	Key Sampling $\mathbf{k}_i$	Key Restoration $\hat{\mathbf{k}}_i$	Used in
greedy-search	$\Xi = \{\xi^1, ..., \xi^K\} \overset{i.i.d}{\sim} \mathcal{U}(\mathbb{R}^L)$ $\mathbf{k} \sim \mathcal{U}(\Xi)$	$\hat{\mathbf{k}} = \arg\max_{\xi \in \Xi} d_{\text{edit}}(\xi, \hat{\mathbf{y}}), \text{ where}$ $d_{\text{edit}}(\xi, \hat{\mathbf{y}}) = \max\{d_{\text{edit}}(\xi_{2:}, \hat{\mathbf{y}}_{2:}) + S(\xi_1, \hat{\mathbf{y}}_1),$ $d_{\text{edit}}(\xi, \hat{\mathbf{y}}_{2:}) - \eta,$ $d_{\text{edit}}(\xi_{2:}, \hat{\mathbf{y}}) - \eta\}$	ITS(Kuditipudi et al., 2023), EXP(Kuditipudi et al., 2023)
context-hash	$\mathbf{k}_i = \text{hash}(\mathbf{y}_{i-c:i-1})$	$\mathbf{k}_i = \text{hash}(\hat{\mathbf{y}}_{i-c:i-1})$	KGW(Kirchenbauer et al., 2023a), Gamma(Hu et al., 2023), Delta(Hu et al., 2023)
fixed-constant	$k \sim \mathcal{U}(\mathbb{R})$; $\mathbf{k}_i = k$	$\mathbf{k}_i = k$	Unigram(Zhao et al., 2023)

Table 5: Key modules of typical watermarks. Both Ξ and k are initialized once, and then fixed on every generation. L is the maximum length of LLM. N, c, η are all fixed hyper-parameters.

A Typical Watermark Designs

Here we list mark and key modules of typical watermarking methods in Table 4 and 5.

B Algorithms of WaterPool

In this section, we present the pseudo codes for EXP-Pool, ITS-Pool and KGW-Pool. We highlight the codes WaterPool invokes in with triangle comments (▷), while the rest of the code remains unchanged from the original watermarking techniques. The invocation code of WaterPool is little, demonstrating the ease of its integration with existing watermarking methods. The specific implementation of modification function F and per-token statistic S used in the pseudo codes can be found in Table 4.

Algorithm 1 EXP-Pool Generation

Params: language model M, max output length L, modification function of EXP $F_{exp}(\cdot,\cdot)$, embedding model $Enc(\cdot)$.
Input: N rounds queries $\{\mathbf{x}^n\}_{n=1}^N$.
Output: N rounds outputs $\{\mathbf{y}^n\}_{n=1}^N$, vector database D.

```
$D \leftarrow \{\}$                                   ▷ Initialize vector database
/* Multi-round queries */
for $n \in \{1, ..., N\}$ do
    Input current round prompt $\mathbf{x}^n$
    $\mathbf{y}^n \leftarrow$ empty string
    $\mathbf{k}^n \sim \mathcal{U}(\mathbb{R}^L)$                     ▷ Sample key
    /* Auto-regressive generation */
    for $i \in 1, ..., L$ do
        $P_i \leftarrow P_M(\cdot|\mathbf{x}^n, \mathbf{y}^n_{<i})$
        $\hat{P}_i \leftarrow F_{exp}(\mathbf{k}^n_i, P_i)$
        $\mathbf{y}^n_i \sim \hat{P}_i$
    end for
    $D \leftarrow D \cup \{(Enc(\mathbf{y}^n), \mathbf{k}^n)\}$   ▷ Store key
    Output current round generation $\mathbf{y}^n$
end for
```

Algorithm 2 EXP-Pool Detection

Params: vector database $D = \{(Enc(\mathbf{y}^n), \mathbf{k}^n)\}_n^N$, embedding model $Enc(\cdot)$, permutation resample times T, edit penalty η, per-token statistic of EXP $S_{exp}(\cdot,\cdot)$.
Input: candidate text $\hat{\mathbf{y}}$
Output: p-value of being watermarked $\hat{p}$

```
/* Aggregation of per-token statistic $s_i$ with edit distance trick */
procedure $d_{\text{edit}}(\mathbf{k}, \mathbf{y})$:
    if len$(\mathbf{k}) = 0$ then
        return $-\eta \cdot$ len$(\mathbf{y})$
    else if len$(\mathbf{y}) = 0$ then
        return $-\eta \cdot$ len$(\mathbf{k})$
    else
        $s_i \leftarrow S_{exp}(\mathbf{k}_1, \mathbf{y}_2)$
        return $\max\{d_{\text{edit}}(\mathbf{k}_{2:}, \mathbf{y}_{2:}) + s_i, d_{\text{edit}}(\mathbf{k}_{2:}, \mathbf{y}) - \eta, d_{\text{edit}}(\mathbf{k}, \mathbf{y}_{2:}) - \eta\}$
    end if
end procedure

$n^* \leftarrow \arg\max_n \text{sim}(Enc(\mathbf{y}^n), Enc(\hat{\mathbf{y}}))$   ▷ Retrieve key from vector database
$\hat{\mathbf{k}} \leftarrow \mathbf{k}^{n^*}$
$\hat{V} \leftarrow d_{edit}(\hat{\mathbf{k}}, \hat{\mathbf{y}})$
/* Permutation test */
for $t \in 1, ..., T$ do
    $\mathbf{k}^t \sim \mathcal{U}(\mathbb{R}^L)$
    $V^t \leftarrow d_{edit}(\mathbf{k}^t, \hat{\mathbf{y}})$
end for
/* Calculate $p$-value */
$\hat{p} \leftarrow \frac{1}{T+1}\left(1 + \sum_t \mathbf{1}_{\hat{V} > V^t}\right)$
```

Algorithm 3 ITS-Pool Generation

Params: language model M, max output length L, modification function of ITS $F_{its}(\cdot,\cdot)$, embedding model $Enc(\cdot)$.

Input: N rounds queries $\{\mathbf{x}^n\}_{n=1}^N$.

Output: N rounds outputs $\{\mathbf{y}^n\}_{n=1}^N$, vector database D.

```
$D \leftarrow \{\}$                                   ▷ Initialize vector database
/* Multi-round queries */
for $n \in \{1, ..., N\}$ do
    Input current round prompt $\mathbf{x}^n$
    $\mathbf{y}^n \leftarrow$ empty string
    $\mathbf{k}^n \sim \mathcal{U}(\mathbb{R}^L)$                          ▷ Sample key
    /* Auto-regressive generation */
    for $i \in 1, ..., L$ do
        $P_i \leftarrow P_M(\cdot|\mathbf{x}^n, \mathbf{y}^n_{<i})$
        $\hat{P}_i \leftarrow F_{its}(\mathbf{k}^n_i, P_i)$
        $\mathbf{y}^n_i \sim \hat{P}_i$
    end for
    $D \leftarrow D \cup \{(Enc(\mathbf{y}^n), \mathbf{k}^n)\}$    ▷ Store key
    Output current round generation $\mathbf{y}^n$
end for
```

Algorithm 4 ITS-Pool Detection

Params: vector database $D = \{(Enc(\mathbf{y}^n), \mathbf{k}^n)\}_n^N$, embedding model $Enc(\cdot)$, permutation resample times T, edit penalty η, per-token statistic of ITS $S_{its}(\cdot,\cdot)$.

Input: candidate text $\hat{\mathbf{y}}$

Output: p-value of being watermarked $\hat{p}$

```
/* Aggregation of per-token statistic $s_i$ with edit distance trick */
procedure $d_{\text{edit}}(\mathbf{k}, \mathbf{y})$:
    if len($\mathbf{k}$) $= 0$ then
        return $-\eta \cdot$ len($\mathbf{y}$)
    else if len($\mathbf{y}$) $= 0$ then
        return $-\eta \cdot$ len($\mathbf{k}$)
    else
        $s_i \leftarrow S_{its}(\mathbf{k}_1, \mathbf{y}_2)$
        return $\max\{d_{\text{edit}}(\mathbf{k}_{2:}, \mathbf{y}_{2:}) + s_i, d_{\text{edit}}(\mathbf{k}_{2:}, \mathbf{y}) - \eta, d_{\text{edit}}(\mathbf{k}, \mathbf{y}_{2:}) - \eta\}$
    end if
end procedure

$n^* \leftarrow \arg\max_n \text{sim}(Enc(\mathbf{y}^n), Enc(\hat{\mathbf{y}}))$    ▷ Retrieve key from vector database
$\hat{\mathbf{k}} \leftarrow \mathbf{k}^{n^*}$
$\hat{V} \leftarrow d_{edit}(\hat{\mathbf{k}}, \hat{\mathbf{y}})$
/* Permutation test */
for $t \in 1, ..., T$ do
    $\mathbf{k}^t \sim \mathcal{U}(\mathbb{R}^L)$
    $V^t \leftarrow d_{edit}(\mathbf{k}^t, \hat{\mathbf{y}})$
end for
/* Calculate $p$-value */
$\hat{p} \leftarrow \frac{1}{T+1}\left(1 + \sum_t \mathbf{1}_{\hat{V} > V^t}\right)$
```

Algorithm 5 KGW-Pool Generation

Params: language model M, max output length L, modification function of KGW $F_{kgw}(\cdot, \cdot)$, embedding model $Enc(\cdot)$.

Input: N rounds queries $\{\mathbf{x}^n\}_{n=1}^N$.

Output: N rounds outputs $\{\mathbf{y}^n\}_{n=1}^N$, vector database D.

```
$D \leftarrow \{\}$                    ▷ Initialize vector database
/* Multi-round queries */
for $n \in \{1, ..., N\}$ do
    Input current round prompt $\mathbf{x}^n$
    $\mathbf{y}^n \leftarrow$ empty string
    $k^n \sim \mathcal{U}(\mathbb{R})$                    ▷ Sample key
    /* Auto-regressive generation */
    for $i \in 1, ..., L$ do
        $P_i \leftarrow P_M(\cdot|\mathbf{x}^n, \mathbf{y}^n_{<i})$
        $\hat{P}_i \leftarrow F_{kgw}(k^n, P_i)$
        $\mathbf{y}^n_i \sim \hat{P}_i$
    end for
    $D \leftarrow D + \{(Enc(\mathbf{y}^n), k^n)\}$    ▷ Store key
    Output current round generation $\mathbf{y}^n$
end for
```

Algorithm 6 KGW-Pool Detection

Params: vector database $D = \{(Enc(\mathbf{y}^n), k^n)\}$, embedding model $Enc(\cdot)$, per-token statistic of KGW $S_{kgw}(\cdot, \cdot)$

Input: candidate text $\hat{\mathbf{y}}$

Output: p-value of being watermarked $\hat{p}$

```
$n^* \leftarrow \arg\max_n \text{sim}(Enc(\mathbf{y}^n), Enc(\hat{\mathbf{y}}))$    ▷
    Retrieve key from vector database
$\hat{k} \leftarrow k^{n^*}$
/* Aggregation of per-token statistic $s_i$ via
    summation */
for $i \in 1, ..., \text{len}(\hat{\mathbf{y}})$ do
    $s_i \leftarrow S_{kgw}(\hat{k}, \hat{\mathbf{y}}_i)$
end for
/* Calculation of $z$-score */
$z \leftarrow \sum_i s_i$
/* Calculation of $p$-value */
$p \leftarrow 1 - \Phi(z)$
```

C Theoretical Proofs

C.1 Proof of Proposition 3.1

Proof. Recall that the imperceptibility is defined as

$$\prod_{i,n} P_M(\mathbf{y}_i^n|\mathbf{x}^n, \mathbf{y}_{<i}^n) = \mathbb{E}_{\mathbf{k}^1,\ldots,\mathbf{k}^N}[\prod_{i,n} F(\mathbf{k}_i^n, P_i^n)(\mathbf{y}_i^n)]$$

Given that (1) $\mathbf{k}^1, ..., \mathbf{k}^N \overset{i.i.d}{\sim} \mathcal{U}(\mathbb{R}^L)$; (2) $P_M(\cdot|\mathbf{x}^n, \mathbf{y}_{<i}^n) = \mathbb{E}_{\mathbf{k}_i\sim\mathcal{U}(\mathbb{R})}[F(\mathbf{k}_i, P_i)]$. We have,

$$\begin{aligned}\text{RHS} &= \mathbb{E}_{\mathbf{k}^1\sim\mathcal{U}(\mathbb{R}^L)}\ldots\mathbb{E}_{\mathbf{k}^N\sim\mathcal{U}(\mathbb{R}^L)}[\prod_{i,n} F(\mathbf{k}_i^n, P_i^n)(\mathbf{y}_i^n)] \\ &= \prod_n \mathbb{E}_{\mathbf{k}^n\sim\mathcal{U}(\mathbb{R}^L)}[\prod_i F(\mathbf{k}_i^n, P_i^n)(\mathbf{y}_i^n)] \\ &= \prod_n (\mathbb{E}_{\mathbf{k}_1^n\sim\mathcal{U}(\mathbb{R})}\ldots\mathbb{E}_{\mathbf{k}_L^n\sim\mathcal{U}(\mathbb{R})}[\prod_i F(\mathbf{k}_i^n, P_i^n)(\mathbf{y}_i^n)]) \\ &= \prod_n (\prod_i \mathbb{E}_{\mathbf{k}_i^n\sim\mathcal{U}(\mathbb{R})}[F(\mathbf{k}_i^n, P_i^n)(\mathbf{y}_i^n)]) \\ &= \prod_{n,i} P_M(\mathbf{y}_i^n|\mathbf{x}^n, \mathbf{y}_{<i}^n) = \text{LHS}\end{aligned}$$

□

C.2 Proof of Proposition 3.2 (EXP-Pool's Imperceptibility)

We first recall the distribution modification process of EXP-Pool. Given a private key $\mathbf{k}_i$ as seed, a standard Gumbel vector $\mathbf{g}_{\mathbf{k}_i} \in \mathbb{R}^{|\Sigma|}$ is sampled. Gumbel-max sampling process on the next token distribution $P_i(\cdot) := P_M(\cdot|\mathbf{x}^n, \mathbf{y}_{<i}^n)$ via $\mathbf{g}_{\mathbf{k}_i}$ samples an output token t^*. The degenerate distribution of t^* is then returned.

Lemma C.1. *The mark module of EXP-Pool satisfies the unbiased condition, i.e.*

$$P_M(\cdot|\mathbf{x}^n, \mathbf{y}_{<i}^n) = \mathbb{E}_{\mathbf{k}_i\sim\mathcal{U}(\mathbb{R})}[F_{exp}(\mathbf{k}_i, P_i)]$$

Proof. For simplicity, we denote $P_M(t|\mathbf{x}^n, \mathbf{y}_{<i}^n)$ as $P_i(t)$ and g_t as the Gumbel variable in $\mathbf{g}_{\mathbf{k}_i}$ corresponding to the token t. Since The lemma holds if and only if for any token $t \in \Sigma$,

$$P_M(t|\mathbf{x}^n, \mathbf{y}_{<i}^n) = \mathbb{E}_{\mathbf{g}_{\mathbf{k}_i,j} \overset{i.i.d}{\sim} \text{Gumbel}(0,1)}[\mathbf{1}_{\log P_i(t)+g_t \geq \log P_i(t') + g_{t'}, \forall t' \in \Sigma}]$$

This equation follows as

$$\begin{aligned}\text{RHS} &= P(\log P_i(t)+g_t \geq \log P_i(t')+g_{t'}, \forall t' \in \Sigma) \\ &= P(\exp(-\exp(-g_{t'})) \leq \exp(-\exp(-g_t))^{P_i(t')/P_i(t)}, \\ &\qquad \forall t' \in \Sigma) \\ &= \int_0^1 P(u_{t'} \leq u_t^{P_i(t')/P_i(t)}, \forall t' \in \Sigma | u_t) p(u_t) du_t \\ &= \int_0^1 \prod_{t'\in\Sigma} P(u_{t'} \leq u_t^{P_i(t')/P_i(t)} | u_t) p(u_t) du_t \\ &= \int_0^1 u_t^{\sum_{t'} P_i(t')/P_i(t)} du_t \\ &= P_i(t) = \text{LHS}\end{aligned}$$

,where $u_t := \exp(-\exp(-g_t))$. We have $u_t \sim \mathcal{U}(0, 1)$, since $g_t \sim \text{Gumbel}(0, 1)$. □

From the unbiased condition and the independent condition of WaterPool, the imperceptibility of EXP-Pool immediately follows according to Proposition 3.1.

C.3 Proof of Proposition 3.2 (ITS-Pool's Imperceptibility)

We first recall the distribution modification process of ITS-Pool. Given a private key $\mathbf{k}_i$ as the random seed, a random permutation $\pi_{\mathbf{k}_i} : \Sigma \to \Sigma$ and a uniform variable $\mathrm{u}_{\mathbf{k}_i} \sim \mathcal{U}([0, 1])$ are sampled. ITS conducts an inverse transform sampling on the permuted distribution P^{perm} via u. The sampled token t is transformed back to t^* via inverse permutation $\pi_{\mathbf{k}_i}^{-1}$. The degenerate distribution of t^* is then returned.

Lemma C.2. *The mark module of ITS satisfies the unbiased condition, i.e.*

$$P_M(\cdot|\mathbf{x}^n, \mathbf{y}_{<i}^n) = \mathbb{E}_{\mathbf{k}_i\sim\mathcal{U}(\mathbb{R})}[F_{its}(\mathbf{k}_i, P_i)]$$

Proof. The lemma follows if that given any permutation π and any output token t^*,

$$P_M(\pi(t^*)|\mathbf{x}^n, \mathbf{y}_{<i}^n) = \mathbb{E}_{\mathrm{u}_{\mathbf{k}_i}\sim\mathcal{U}([0,1])}[\mathbf{1}_{\pi(t^*)\text{ is sampled via inverse transform sampling}}]$$

This equation certainly holds because of the definition of inverse transform sampling, i.e.

$$\begin{aligned}\text{RHS} &= P(\mathrm{u}_{\mathbf{k}_i} \in [P_M(\{t' : \text{ord}(t') < \text{ord} \circ \pi(t^*)\}|\mathbf{x}^n, \mathbf{y}_{<i}^n), \\ &\qquad P_M(\{t' : \text{ord}(t') \leq \text{ord} \circ \pi(t^*)\}|\mathbf{x}^n, \mathbf{y}_{<i}^n)\,]\,) \\ &= P_M(\pi(t^*)|\mathbf{x}^n, \mathbf{y}_{<i}^n) = \text{LHS}\end{aligned}$$

,where $ord : \Sigma \to |\Sigma|$ is a function maps each token to its order in vocabulary. □

From the unbiased condition and the independent condition of WaterPool, the imperceptibility of ITS-Pool immediately follows according to Proposition 3.1.

C.4 Proof of Proposition 3.3 (EXP-Pool's Efficacy)

We first recall the mark module of EXP-Pool (*gumbel-sample* in Table 4). During generation, the modification function $F_{exp}(\mathbf{k}_i, P_i)$ takes a private key $\mathbf{k}_i$ as seed to sample a standard Gumbel vector. A Gumbel-max sampling is then conduct to sample an output token, of which the degenerate distribution is returned. During detection, the mark module takes in a restored private key $\hat{\mathbf{k}}_i$ to generate a restored Gumbel vector $\mathbf{g}_{\hat{\mathbf{k}}_i} \in \mathbb{R}^{|\Sigma|}, \mathbf{g}_{\hat{\mathbf{k}}_i,j} \sim \text{Gumbel}(0,1)$. The per-token statistic $S_{exp}(\hat{\mathbf{k}}_i, \mathbf{y}_i) = -\exp(-\mathbf{g}_{\hat{\mathbf{k}}_i,\text{ord}(\mathbf{y}_i)})$ is then calculated, where $\text{ord} : \Sigma \to \{1, ..., |\Sigma|\}$ is a function maps each token to the its order in the vocabulary.

For simplicity, we denote g_t as the Gumbel variable $\mathbf{g}_{\mathbf{k}_i,\text{ord}(t)}$ corresponding to the token t.

We begin by proving that given a candidate token $\mathbf{y}_i$, the expectation of per-token statistic only relies on the original distribution $P_i(\mathbf{y}_i)$ if $\mathbf{y}_i$ is sampled from the modified distribution seeded by $\mathbf{k}_i$. It can be formalized as the following lemma.

Lemma C.3. *Given a prefix $\mathbf{y}_{<i}$ and a token $\mathbf{y}_i$, for a private key $\mathbf{k}_i \sim \mathcal{U}(\mathbb{R})$, if $\mathbf{y}_i$ is sampled from $F_{exp}(\mathbf{k}_i, P_M(\cdot|\mathbf{y}_{<i}))$, then $\mathbb{E}_{\mathbf{k}_i \sim \mathbb{R}}[S_{exp}(\mathbf{k}_i, \mathbf{y}_i)|\mathbf{y}_i, \mathbf{y}_{<i}] = -P_i(\mathbf{y}_i)$.*

Proof. The randomness of $\mathbf{k}_i$ affects the statistic via the Gumbel vector $\mathbf{g}_{\mathbf{k}_i} = [g_t]_{t\in\Sigma}$. For simplicity, we only consider the randomness of $\mathbf{g}_{\mathbf{k}_i}$ instead of $\mathbf{k}_i$.

We first calculate the cumulative distribution function of $S_{exp}(\mathbf{k}_i, \mathbf{y}_i)|\mathbf{y}_i, \mathbf{y}_{<i}$.

$$
\begin{aligned}
&P(S_{exp}(\mathbf{k}_i, \mathbf{y}_i) \leq v|\mathbf{y}_i, \mathbf{y}_{<i}) \\
=& P(g_{\mathbf{y}_i} \leq -\log(-v)|\mathbf{y}_i, \mathbf{y}_{<i}) \\
\overset{(1)}{=}& P(\bigcap_{t\in\Sigma} g_t + \log P_i(t) \leq g_{\mathbf{y}_i} + \log P_i(\mathbf{y}_i) \\
&\qquad \leq -\log(-v) + \log P_i(\mathbf{y}_i) \mid \mathbf{y}_i, \mathbf{y}_{<i}) \\
\overset{(2)}{=}& \prod_{t\in\Sigma} \exp(-v\frac{P_i(t)}{P_i(\mathbf{y}_i)}) \\
=& \exp(-v/P_i(\mathbf{y}_i))
\end{aligned}
$$

, where the equation (1) follows from the definition of Gumbel max sampling; (2) follows from $g_t \sim \text{Gumbel}(0,1)$. Therefore, $-S_{exp}(\mathbf{k}_i, \mathbf{y}_i)|\mathbf{y}_i, \mathbf{y}_{<i} \sim Exp(1/P_i(\mathbf{y}_i))$. The lemma follows immediately by calculating the expectation. □

On the contrary, if $\mathbf{y}_i$ is not sampled from modified distribution seeded by $\mathbf{k}_i$,

$$
P(S_{exp}(\mathbf{k}_i,\mathbf{y}_i) \leq v|\mathbf{y}_i,\mathbf{y}_{<i}) = P(g_{\mathbf{y}_i} \leq -\log(-v)) = \exp(v)
$$

from which we have

$$
\begin{aligned}
-S_{exp}(\mathbf{k}_i, \mathbf{y}_i)|\mathbf{y}_i, \mathbf{y}_{<i} &\sim Exp(1) \\
\mathbb{E}_{\mathbf{k}_i\sim\mathcal{U}(\mathbb{R})}[S_{exp}(\mathbf{k}_i, \mathbf{y}_i)|\mathbf{y}_i, \mathbf{y}_{<i}] &= -1
\end{aligned}
$$

Eventually, we can guaranteed the statistical difference of EXP-Pool given the prefix $\mathbf{y}_{<i}$,

$$
\begin{aligned}
&\mathbb{E}[S_{exp}(\mathbf{k}_i,\mathbf{y}_i)|\mathbf{y}_{<i},H_1] - \mathbb{E}[S_{exp}(\mathbf{k}_i,\mathbf{y}_i)|\mathbf{y}_{<i},H_0] \\
=&\mathbb{E}_{\mathbf{y}_i,\mathbf{k}_i}[S_{exp}(\hat{\mathbf{k}}_i,\mathbf{y}_i)|\mathbf{y}_{<i},H_1,\hat{\mathbf{k}}_i=\mathbf{k}_i]\cdot p_{recall} \\
&+\mathbb{E}_{\mathbf{y}_i,\mathbf{k}_i}[S_{exp}(\hat{\mathbf{k}}_i,\mathbf{y}_i)|\mathbf{y}_{<i},H_1,\hat{\mathbf{k}}_i\neq\mathbf{k}_i]\cdot(1-p_{recall})+1 \\
=&\mathbb{E}_{\mathbf{y}_i}[-P_i(\mathbf{y}_i)|\mathbf{y}_{<i}]\cdot p_{recall}-(1-p_{recall})+1 \\
=&\sum_{\mathbf{y}_i\in\Sigma}(1-P_i(\mathbf{y}_i))P_i(\mathbf{y}_i)\cdot p_{recall} := \phi_{exp}(\boldsymbol{p}^i)\cdot p_{recall}
\end{aligned}
$$

, where $\phi_{exp}(\boldsymbol{p}^i)$ is only relevant to probability vector $\boldsymbol{p}^i$ of $P_M(\cdot|\mathbf{y}_{<i})$, representing watermarking potentials, and p_{recall} is the recall of the retriever in EXP-Pool.

C.5 Proof of Proposition 3.3 (ITS-Pool's Efficacy)

We first recall the mark module of ITS-Pool (*inverse-sample* in Table 4). During generation, the modification function $F_{its}(\mathbf{k}_i, P_i)$ takes a private key $\mathbf{k}_i$ as seed to sample a standard uniform variable and a random permutation. An inverse transform sampling is then conduct on the permuted distribution to sample an output token, of which the degenerate distribution is returned. During detection, the mark module takes in a restored private key $\hat{\mathbf{k}}_i$ to restore a standard uniform variable $\text{u}_{\hat{\mathbf{k}}_i} \sim \mathcal{U}(0,1)$ and a random permutation $\pi_{\hat{\mathbf{k}}_i} : \Sigma \to \Sigma$. The per-token statistic $S_{its}(\hat{\mathbf{k}}_i, \mathbf{y}_i) = (\text{u}_{\hat{\mathbf{k}}_i} - \frac{1}{2})(\frac{\text{ord}(\pi_{\hat{\mathbf{k}}_i}(\mathbf{y}_i))-1}{|\Sigma|-1} - \frac{1}{2})$ is then calculated, where $\text{ord} : \Sigma \to \{1, ..., |\Sigma|\}$ is a function maps each token to the its order in the vocabulary.

We begin by proving that given a candidate token $\mathbf{y}_i$, the expectation of per-token statistic only relies on the original distribution $P_i(\mathbf{y}_i)$ if $\mathbf{y}_i$ is sampled from the modified distribution seeded by $\mathbf{k}_i$ (i.e. the alternative hypothesis H_1). It can be formalized as the following lemma[11].

Lemma C.4. *Given a prefix $\mathbf{y}_{<i}$ and a token $\mathbf{y}_i$, for a private key $\mathbf{k}_i \sim \mathcal{U}(\mathbb{R})$, if $\mathbf{y}_i$ is sampled from $F_{its}(\mathbf{k}_i, P_M(\cdot|\mathbf{y}_{<i}))$, then $\mathbb{E}_{\mathbf{k}_i\sim\mathcal{U}(\mathbb{R})}[S_{its}(\mathbf{k}_i, \mathbf{y}_i)|\mathbf{y}_i, \mathbf{y}_{<i}] = C_0 \cdot (1-P_i(\mathbf{y}_i))$, where C_0 is a constant relevant to vocabulary size $|\Sigma|$.*

Proof. The randomness of $\mathbf{k}_i$ affects the statistic via the uniform variable $\text{u}_{\mathbf{k}_i}$ and the permutation $\pi_{\mathbf{k}_i}$. For simplicity, we omit the subscript $\mathbf{k}_i$ and only consider the randomness of u and π.

[11] It is based on Lemma B.1 in (Kuditipudi et al., 2023)

We show that $\pi|\mathbf{y}_i, \mathbf{y}_{<i}$ is a uniform random variable over the permutation space.

$$P(\pi|\mathbf{y}_i, \mathbf{y}_{<i}) = \frac{P(\mathbf{y}_i|\pi, \mathbf{y}_{<i})P(\pi)}{P(\mathbf{y}_i|\mathbf{y}_{<i})} = P(\pi)$$

, where the second equation follows from that permutation won't affect the inverse transform sampling (see AppendixC.3)

We also prove that $\mathrm{u}|\pi, \mathbf{y}_i, \mathbf{y}_{<i}$ is a uniform random variable. We define the interval of $\mathbf{y}_i$ given π in inverse transform sampling during ITS generation,

$$I(\mathbf{y}_i,\pi)=[P_i(\{t':\mathrm{ord}(t')<\mathrm{ord}\circ\pi(\mathbf{y}_i)\}), \\ P_i(\{t':\mathrm{ord}(t')\leq\mathrm{ord}\circ\pi(\mathbf{y}_i)\})]$$

It is evident that $|I(\mathbf{y}_i, \pi) = P_i(\mathbf{y}_i)|$. Then for any interval $I \subset [0, 1]$ we have

$$P(\mathrm{u} \in I|\mathbf{y}_i, \pi, \mathbf{y}_{<i}) = \frac{P(\mathbf{y}_i, \mathrm{u} \in I|\pi, \mathbf{y}_{<i})}{P(\mathbf{y}_i|\pi, \mathbf{y}_{<i})} = \frac{|I \cap I(\mathbf{y}_i, \pi)|}{|I(\mathbf{y}_i, \pi)|}$$

So $\mathrm{u}|\pi, \mathbf{y}_i, \mathbf{y}_{<i} \sim \mathcal{U}(I(\mathbf{y}_i, \pi))$. Then we have,

$$\begin{aligned}
&\mathbb{E}[\mathrm{u}|\mathbf{y}_i,\pi(\mathbf{y}_i),\mathbf{y}_{<i}] \\
=&\mathbb{E}\left[P_i(\{t':\mathrm{ord}(t')<\mathrm{ord}\circ\pi(\mathbf{y}_i)\})+\frac{|I(\mathbf{y}_i,\pi)|}{2}\middle|\mathbf{y}_i,\pi(\mathbf{y}_i),\mathbf{y}_{<i}\right] \\
=&\frac{(\pi(\mathbf{y}_i)-1)}{|\Sigma|-1}\cdot(1-P_i(\mathbf{y}_i))+\frac{P_i(\mathbf{y}_i)}{2} \\
=&\frac{1}{2}+(\frac{(\pi(\mathbf{y}_i)-1)}{|\Sigma|-1}-\frac{1}{2})(1-P_i(\mathbf{y}_i))
\end{aligned}$$

It is evident that $\mathbb{E}[\mathrm{u}] = \frac{1}{2}$ and $\mathbb{E}[\frac{(\pi(\mathbf{y}_i)-1)}{|\Sigma|-1}] = \frac{1}{2}$, since they are both uniform standard variables. Therefore, $S_{its}(\mathbf{k}_i, \mathbf{y}_i)$ essentially calculates the covariance between u and $\frac{\pi(\mathbf{y}_i)-1}{|\Sigma|-1}$, which is tractable as following,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{k}_i}[S_{its}(\mathbf{k}_i,\mathbf{y}_i)|\mathbf{y}_i,\mathbf{y}_{<i}] \\
=&\mathrm{Cov}\left(\mathrm{u},\frac{\pi(\mathbf{y}_i)-1}{|\Sigma|-1}\middle|\mathbf{y}_i,\mathbf{y}_{<i}\right) \\
=&(\mathrm{u}-\frac{1}{2})(\frac{\pi(\mathbf{y}_i)-1}{|\Sigma|-1}-\frac{1}{2})\cdot P(\mathrm{u},\pi(\mathbf{y}_i)|\mathbf{y}_i,\mathbf{y}_{<i}) \\
=&\mathbb{E}[\mathrm{u}-\frac{1}{2}|\mathbf{y}_i,\pi(\mathbf{y}_i),\mathbf{y}_{<i}]\cdot(\frac{\pi(\mathbf{y}_i)-1}{|\Sigma|-1}-\frac{1}{2})\cdot P(\pi(\mathbf{y}_i)|\mathbf{y}_i,\mathbf{y}_{<i}) \\
=&(1-P_i(\mathbf{y}_i))\cdot(\frac{\pi(\mathbf{y}_i)-1}{|\Sigma|-1}-\frac{1}{2})^2\cdot P(\pi(\mathbf{y}_i)|\mathbf{y}_i,\mathbf{y}_{<i}) \\
=&(1-P_i(\mathbf{y}_i))\cdot\mathrm{Var}\left(\frac{\pi(\mathbf{y}_i)-1}{|\Sigma|-1}\middle|\mathbf{y}_i,\mathbf{y}_{<i}\right) \\
=&C_0\cdot(1-P_i(\mathbf{y}_i))
\end{aligned}$$

,where $C_0 = \mathrm{Var}\left(\frac{\pi(\mathbf{y}_i)-1}{|\Sigma|-1}\middle| \mathbf{y}_i, \mathbf{y}_{<i}\right)$ is a constant since $\pi|\mathbf{y}_i, \mathbf{y}_{<i}$ is uniform over the space of vocabulary permutation. □

On the contrary, if $\mathbf{y}_i$ is not sampled from modified distribution seeded by $\mathbf{k}_i$, $\mathbb{E}_{\mathbf{k}_i}[S_{its}(\mathbf{k}_i, \mathbf{y}_i)|\mathbf{y}_i, \mathbf{y}_{<i}]$ =Cov(u, $\frac{\pi(\mathbf{y}_i)-1}{|\Sigma|-1}|\mathbf{y}_i, \mathbf{y}_{<i})$ still holds. Now that $\mathbf{k}_i$ and $\mathbf{y}_i$ are independent, $\mathbb{E}_{\mathbf{k}_i}[S_{its}(\mathbf{k}_i, \mathbf{y}_i)|\mathbf{y}_i, \mathbf{y}_{<i}] = 0$ trivially. Therefore, $\mathbb{E}_{\mathbf{y}_i,\mathbf{k}_i}[S_{its}(\hat{\mathbf{k}}_i, \mathbf{y}_i)|\mathbf{y}_{<i}, H_0] = 0$ follows immediately.

Under the alternative hypothesis H_1, the lemma above provides that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{y}_i,\mathbf{k}_i}[S_{its}(\hat{\mathbf{k}}_i, \mathbf{y}_i)|\mathbf{y}_{<i}, H_1] \\
&= \mathbb{E}_{\mathbf{y}_i,\mathbf{k}_i}[S_{its}(\hat{\mathbf{k}}_i, \mathbf{y}_i)|\mathbf{y}_{<i}, H_1, \hat{\mathbf{k}}_i = \mathbf{k}_i] \cdot p_{recall} \\
&\quad + \mathbb{E}_{\mathbf{y}_i,\mathbf{k}_i}[S_{its}(\hat{\mathbf{k}}_i, \mathbf{y}_i)|\mathbf{y}_{<i}, H_1, \hat{\mathbf{k}}_i \neq \mathbf{k}_i] \cdot (1 - p_{recall}) \\
&= \mathbb{E}_{\mathbf{y}_i}[C_0 \cdot (1 - P_i(\mathbf{y}_i))|\mathbf{y}_{<i}] \cdot p_{recall} \\
&= C_0 \cdot p_{recall} \cdot \sum_{\mathbf{y}_i\in\Sigma} (1 - P_i(\mathbf{y}_i))P_i(\mathbf{y}_i)
\end{aligned}$$

, where p_{recall} represents the recall performance of the retriever in ITS-Pool.

Finally, we can guarantee the statistical difference of ITS-Pool,

$$\begin{aligned}
&\mathbb{E}[S_{its}(\mathbf{k}_i, \mathbf{y}_i)|\mathbf{y}_{<i}, H_1] - \mathbb{E}[S_{its}(\mathbf{k}_i, \mathbf{y}_i)|\mathbf{y}_{<i}, H_0] \\
&= C_0\cdot \sum_{\mathbf{y}_i\in\Sigma} (1-P_i(\mathbf{y}_i))P_i(\mathbf{y}_i)\cdot p_{recall} := \phi_{its}(\boldsymbol{p}^i)\cdot p_{recall}
\end{aligned}$$

, where $\phi_{its}(\boldsymbol{p}^i)$ is only relevant to probability vector $\boldsymbol{p}^i$ of $P_M(\cdot|\mathbf{y}_{<i})$, representing watermarking potentials.

C.6 Proof of Proposition 3.3 (KGW-Pool's Efficacy)

We first recall the mark module of KGW-Pool (*logits-add* in Table 4). During generation, the mark module will randomly sample a green list $\mathcal{G}_{\mathbf{k}_i}$ of $\gamma|\Sigma|$ tokens from vocabulary, which is seeded by $\mathbf{k}_i$. Logits of these green tokens are increased by a constant δ to form the modified distribution $F_{kgw}(\mathbf{k}_i, P_i(\mathbf{y}_i))$. During detection, the mark module takes in a restored private key $\hat{\mathbf{k}}_i$, generates a green list $\mathcal{G}_{\hat{\mathbf{k}}_i}$ seeded by $\hat{\mathbf{k}}_i$, and then calculates the per-token statistic $S_{kgw}(\hat{\mathbf{k}}_i, \mathbf{y}_i) = \frac{\mathbf{1}_{\mathbf{y}_i\in\mathcal{G}_{\hat{\mathbf{k}}_i}} - \gamma}{\sqrt{\mathrm{len}(\mathbf{y})\gamma(1-\gamma)}}$.

For simplicity, we omit the subscript $\mathbf{k}_i$ in $\mathcal{G}_{\mathbf{k}_i}$. We also denote the size of vocabulary and green list as $N = |\Sigma|$ and $N_G = \gamma|\Sigma|$ respectively.

Since the denominator is a constant under both hypotheses, we only need to focus on $S'(\mathbf{k}_i, \mathbf{y}_i) := \mathbf{1}_{\mathbf{y}_i\in\mathcal{G}_{\mathbf{k}_i}}$. Under the alternative hypothesis, the expectation of S' is essentially the probability of sampling a token from the green list during KGW-Pool generation. We show that the probability can be bounded from below, as formalized in the following lemma[12].

[12] It is based on Lemma F.1 in (Kirchenbauer et al., 2023a)

Lemma C.5. *Given a prefix $\mathbf{y}_{<i}$, for a private key $\mathbf{k}_i \sim \mathcal{U}(\mathbb{R})$ and a token $\mathbf{y}_i$, if $\mathbf{y}_i$ is sampled from $F_{kgw}(\mathbf{k}_i, P_M(\cdot|\mathbf{y}_{<i}))$, then $\mathbb{E}_{\mathbf{k}_i,\mathbf{y}_i}[S'(\mathbf{k}_i,\mathbf{y}_i)|\mathbf{y}_{<i}] \geq C_1 \cdot \text{Spike}(\boldsymbol{p}^i, \frac{(1-\gamma)(\alpha-1)}{1+(\alpha-1)\gamma})$, where C_1 is a constant and* Spike$(\boldsymbol{p}^i, c)$ *is the spike entropy defined in (Kirchenbauer et al., 2023a).*

Proof. Trivially, we have $\mathbb{E}_{\mathbf{k}_i,\mathbf{y}_i}[S'(\mathbf{k}_i,\mathbf{y}_i)|\mathbf{y}_{<i}] = P(\mathbf{y}_i \in \mathcal{G}, \mathbf{y}_i, \mathcal{G}|\mathbf{y}_{<i})$. We consider the following process of sampling $\mathbf{y}_i$ and $\mathcal{G}$. We first randomly choose a token t as output token $\mathbf{y}_i$, and then randomly sample the remaining tokens to construct the green list. Therefore, the expected probability of a token from the green list being sampled can be written as,

$$\mathbb{E}_{\mathbf{y}_i\in\Sigma}\mathbb{E}_{\mathcal{G}\ s.t. \mathbf{y}_i\in\mathcal{G}}\frac{\alpha P_i(\mathbf{y}_i)}{\sum_{t\in\Sigma}P_i(t)+\alpha\sum_{t\in\mathcal{G}}P_i(t)}$$

, where $\alpha = \exp(\delta)$.

Define the inner expectation as $f_{\mathbf{y}_i}(\boldsymbol{p}^i)$, where $\boldsymbol{p}^i$ is the probability vector of $P_i(\cdot)$. Trivially, $f_{\mathbf{y}_i}(\boldsymbol{p}^i) = f_{\mathbf{y}_i}(\Pi\boldsymbol{p}^i)$ for any permutation Π over the vocabulary except $\mathbf{y}_i$. Also, $f_{\mathbf{y}_i}$ is convex in $\boldsymbol{p}^i_{-\mathbf{y}_i}$. Therefore, we have,

$$\begin{aligned}
&f_{\mathbf{y}_i}(\boldsymbol{p}^i) = \mathbb{E}_\Pi f_{\mathbf{y}_i}(\Pi\boldsymbol{p}^i) \overset{(1)}{\geq} f_{\mathbf{y}_i}(\mathbb{E}_\Pi\Pi\boldsymbol{p}^i)\\
&\overset{(2)}{\geq} \alpha P_i(\mathbf{y}_i)/((1-P_i(\mathbf{y}_i))(N-N_G)/(N-1)\\
&\quad+\alpha(1-P_i(\mathbf{y}_i))(N_G-1)/(N-1)+\alpha P_i(\mathbf{y}_i))\\
&= P_i(\mathbf{y}_i)\frac{\alpha N-\alpha}{N-N_G+\alpha N_G+(\alpha-1)(N-N_G)P_i(\mathbf{y}_i)-\alpha}\\
&\geq P_i(\mathbf{y}_i)\frac{\alpha N}{N-N_G+\alpha N_G+(\alpha-1)(N-N_G)P_i(\mathbf{y}_i)}\\
&= \frac{\alpha P_i(\mathbf{y}_i)}{(1-\gamma)+\alpha\gamma+(\alpha-1)(1-\gamma)P_i(\mathbf{y}_i)}
\end{aligned}$$

, where (1) follows from Jensen's inequality; (2) follows from $\mathbb{E}_\Pi\Pi\boldsymbol{p}^i_t = \frac{1-P_i(\mathbf{y}_i)}{N-1}$, $\forall t \neq \mathbf{y}_i$. Then we have,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{k}_i\mathbf{y}_i}[S'(\mathbf{k}_i,\mathbf{y}_i)|\mathbf{y}_{<i}]\\
&= P(\mathbf{y}_i \in \mathcal{G}, \mathbf{y}_i, \mathcal{G}|\mathbf{y}_{<i})\\
&= N_G \cdot \mathbb{E}_{\mathbf{y}_i\in\Sigma}\mathbb{E}_{\mathcal{G}\ s.t. \mathbf{y}_i\in\mathcal{G}}\frac{\alpha P_i(\mathbf{y}_i)}{\sum_{t\in\Sigma}P_i(t)+\alpha\sum_{t\in\mathcal{G}}P_i(t)}\\
&= N_G \cdot \mathbb{E}_{\mathbf{y}_i\in\Sigma}f_{\mathbf{y}_i}(\boldsymbol{p}^i)\\
&\geq \frac{\gamma\alpha}{1+(\alpha-1)\gamma}\text{Spike}(\boldsymbol{p}^i, \frac{(1-\gamma)(\alpha-1)}{1+(\alpha-1)\gamma})\\
&:= C_1 \cdot \text{Spike}(\boldsymbol{p}^i, \frac{(1-\gamma)(\alpha-1)}{1+(\alpha-1)\gamma})
\end{aligned}$$

, where Spike$(\boldsymbol{p}^i, c) = \sum_{t\in\Sigma}\frac{P_i(t)}{1+cP_i(t)}$. And the lower bound is strictly larger than γ. □

On the contrary, under the null hypothesis, trivially we have $\mathbb{E}[S'(\mathbf{k}_i,\mathbf{y}_i)|\mathbf{y}_{<i}, H_0] = \gamma$, since $\mathbf{k}_i$ and $\mathbf{y}_i$ are independent. Combining all above, we eventually have,

$$\begin{aligned}
&\mathbb{E}[S(\hat{\mathbf{k}}_i,\mathbf{y}_i)|\mathbf{y}_{<i},H_1]-\mathbb{E}[S(\hat{\mathbf{k}}_i,\mathbf{y}_i)|\mathbf{y}_{<i},H_0]\\
&=\mathbb{E}_{\mathbf{k}_i,\mathbf{y}_i}[S(\hat{\mathbf{k}}_i,\mathbf{y}_i)|\mathbf{y}_{<i},H_1,\hat{\mathbf{k}}_i=\mathbf{k}_i]\cdot p_{rec}\\
&\quad+\mathbb{E}_{\mathbf{k}_i,\mathbf{y}_i}[S(\hat{\mathbf{k}}_i,\mathbf{y}_i)|\mathbf{y}_{<i},H_1,\hat{\mathbf{k}}_i\neq\mathbf{k}_i]\cdot(1-p_{rec})\\
&=(\mathbb{E}_{\mathbf{k}_i,\mathbf{y}_i}[S'(\mathbf{k}_i,\mathbf{y}_i)|\mathbf{y}_{<i},H_1]-\gamma)/\sqrt{\text{len}(\mathbf{y})\gamma(1-\gamma)}\cdot p_{rec}\\
&\geq(C_1\cdot\text{Spike}(\boldsymbol{p}^i,\frac{(1-\gamma)(\alpha-1)}{1+(\alpha-1)\gamma})-\gamma)/\sqrt{\text{len}(\mathbf{y})\gamma(1-\gamma)}\cdot p_{rec}\\
&:=\phi_{kgw}(\boldsymbol{p}^i)\cdot p_{rec}
\end{aligned}$$

, where $\phi_{kgw}(\boldsymbol{p}^i)$ is only relevant to probability vector $\boldsymbol{p}^i$ of $P_M(\cdot|\mathbf{y}_{<i})$, representing watermarking potentials at this step, and p_{rec} is the recall performance of the retriever in KGW-Pool.

D Experimental Details

Datasets. Following previous works (Kirchenbauer et al., 2023a,b), we include two common used datasets for our experiments, the Colossal Common Crawl Cleaned corpus (C4) and "Explain Like I'm Five" (ELI5) (Fan et al., 2019). C4 is a colossal, cleaned version of Common Crawl's web crawl corpus[13], which has been commonly adopted as pretraining corpus of LLMs. We randomly select 3000 texts of length 50 from C4 as prompts for open-ended generation task. ELI5 is a dataset of questions and answers gathered from three topics of reddits, where users ask factual questions requiring paragraph-length or longer answers. Specifically, we use the version curated by (Krishna et al., 2023) including 2758 samples. The human-written questions are used as prompts for long-form question answering.

Metrics. We generate 20 watermarked outputs for each prompt, while considering outputs of the original LLM as non-watermarked. Subsequently, a total of about 120,000 samples are used to evaluate each watermarking technique. For both efficacy and robustness, we report true positive rate (watermarked texts being successfully detected) at 1% false positive rate (non-watermarked texts being falsely detected) denoted as TPR@FPR=1%. To comprehensively evaluate the robustness of watermarking techniques, we include three different kinds of attacks, namely Lexical-Attack, Dipper-Attack and Translation-Attack. Lexical-attack is a baseline attack by randomly add/delete/replace a small portion of texts. Specifically, we randomly modify 10% tokens of the original wa-

[13] https://commoncrawl.org

termarked text. Dipper is a paraphrasing model proposed by (Krishna et al., 2023) aiming at corrupting watermark patterns within texts. We set its hyper-parameters `lex=40,div=40` following (Kirchenbauer et al., 2023b). Translation-attack represents roundtrip-translation, which is a widely used paraphrasing method. Following (Kuditipudi et al., 2023), we translate texts to Russian and then translate them back to English. For the evaluation of imperceptibility, we split the criteria into two aspects: (1) the distribution bias within each output (2) the independence among different outputs. The former can be evaluated with perplexity while the latter can be roughly evaluated with n-gram distinction (Kirchenbauer et al., 2023b). Specifically, we consider the distinction across all outputs (Glob-distinct-N) and within outputs in response to one single prompt (Group-distinct-N).

Baselines. We include several typical methods as baselines. In addition to EXP(Kuditipudi et al., 2023), ITS(Kuditipudi et al., 2023) and KGW(Kirchenbauer et al., 2023a), we also include Gamma(Hu et al., 2023), Delta(Hu et al., 2023) and Unigram(Zhao et al., 2023). Both Gamma and Delta propose unbiased modification functions towards optimal imperceptibility during generation, and conduct likelihood ratio test for detection. Unigram is similar to KGW, fixing the private key during generation, leading to a fixed green list partition in sequence level.

Implementation details. We conduct main experiments on two LLMs of different scales, OPT-1.3b and OPT-6.7b, following (Krishna et al., 2023). OPT-1.3b is used by default except in main experiments. On both open-ended generation and long-form question answering, we conduct multinomial sampling to generate sequences within the range of [50, 70] tokens. We use a 128 dimension sentence embedding model (Nussbaum et al., 2024) as the retriever in WaterPool. As for implementation of mark modules in different WaterPool (i.e. KGW-Pool, ITS-Pool, EXP-Pool), we use identical hyper-parameter settings as the original watermarking technique. All baselines are reproduced based on source codes provided by original paper. For KGW, we set $\delta = 2.0$, $\gamma = 0.25$ as suggested in (Kirchenbauer et al., 2023a). For Unigram, we set $\delta = 2.0$, $\gamma = 0.5$ as suggested in (Zhao et al., 2023). For Gamma and Delta, we use the context length of 5 and search the perturbation strength d over the set$\{0, 0.1, ..., 1.0\}$ following (Christ et al., 2023). For EXP and ITS, we set the key length $n = 80$, large enough to generate at most 70 tokens in our main experiments. We set the edit-distance penalty $\gamma = 0.0$ and 0.4 respectively following (Kuditipudi et al., 2023). For KGW-Pool, we observed high variance of performance. It is because of the random partition in a sequence level. The logits-add mark module is sensitive to this partition, which is similar to Unigram. To this end, we resample the key for three times and use the best one as watermarked output during generation in practice. This trick only leads to additional time complexity of generation and won't affect any other analysis in this paper. EXP, ITS, EXP-Pool and ITS-Pool all leverage permutation tests to calculate p-value (e.g. `/* Permutation test */` in Algorithm 2). We conduct the permutations with 5000 resamples. Following (Kuditipudi et al., 2023), we only pre-compute the permutation distribution once instead of recomputing it for each candidate text. This trick reduces the high time complexity of permutation tests and doesn't cause much performance degradation.

E Additional Experiments

E.1 Full Results of WaterPool

In this section, we present full results of WaterPool under different settings (e.g. OPT-1.3b/OPT-6.7b, open-ended generation / long-form question answering) in Table 6, 7, 8 and 9.

E.2 Problem of Retrieval Watermark

In this section, we conduct an experiment to empirically demonstrate the statements in Section 3.4 about the semantic collision problem of the retrieval watermark (Krishna et al., 2023) in real-world scenarios. We hypothesize that retrieval watermark may suffer from severe degradation when facing different non-watermarked text distributions. Therefore, in addition to OPT-1.3b and OPT-6.7b used in main experiments, we include another eight prominent LLMs: Gemma-2b(Team et al., 2024), Gemma-7b(Team et al., 2024), Llama2-7b(Touvron et al., 2023), Llama2-13b(Touvron et al., 2023), Vicuna-7b(Chiang et al., 2023), Vicuna-13b(Chiang et al., 2023), generating outputs for both open-ended generation and long-form question answering tasks. We employ a cross-testing experimental setup, where each time one model is treated as the watermarked model, while the others are considered as non-watermarked models. This setup reflects a common real-world sce-

	w/o Attack		Lexical-Attack		Dipper-Attack		Translation-Attack	
	value↑	Δ	value↑	Δ	value↑	Δ	value↑	Δ
Open Text Generation								
Gamma	$99.58_{\pm 0.01}$	-	$78.69_{\pm 0.08}$	-	$55.18_{\pm 0.11}$	-	$58.33_{\pm 0.18}$	-
Delta	$94.24_{\pm 0.05}$	-	$62.48_{\pm 0.29}$	-	$52.47_{\pm 0.10}$	-	$54.52_{\pm 0.16}$	-
Unigram	$99.59_{\pm 0.06}$	-	$99.31_{\pm 0.22}$	-	$\underline{83.60_{\pm 4.75}}$	-	$\underline{90.99_{\pm 2.47}}$	-
KGW	$\underline{99.87_{\pm 0.01}}$	-	$99.24_{\pm 0.02}$	-	$77.43_{\pm 0.48}$	-	$85.88_{\pm 0.06}$	-
KGW-Pool	$\mathbf{99.90}_{\pm 0.00}$	$0.03_{\pm 0.01}$	$\mathbf{99.74}_{\pm 0.00}$	$0.50_{\pm 0.02}$	$\mathbf{84.10}_{\pm 0.89}$	$6.67_{\pm 1.37}$	$\mathbf{92.15}_{\pm 0.14}$	$6.27_{\pm 0.19}$
EXP	$99.45_{\pm 0.02}$	-	$98.97_{\pm 0.02}$	-	$73.60_{\pm 0.49}$	-	$80.45_{\pm 0.05}$	-
EXP-Pool	$99.76_{\pm 0.01}$	$0.31_{\pm 0.02}$	$\underline{99.57_{\pm 0.00}}$	$0.60_{\pm 0.02}$	$80.42_{\pm 0.94}$	$6.83_{\pm 1.18}$	$90.65_{\pm 0.04}$	$10.20_{\pm 0.09}$
ITS	$95.48_{\pm 0.03}$	-	$83.19_{\pm 0.11}$	-	$59.24_{\pm 0.13}$	-	$56.88_{\pm 0.03}$	-
ITS-Pool	$99.18_{\pm 0.01}$	$3.70_{\pm 0.03}$	$96.70_{\pm 0.00}$	$13.52_{\pm 0.11}$	$61.68_{\pm 0.03}$	$2.44_{\pm 0.15}$	$71.33_{\pm 0.03}$	$14.45_{\pm 0.06}$
Long-Form Question Answering								
Gamma	$99.85_{\pm 0.01}$	-	$80.93_{\pm 0.17}$	-	$55.14_{\pm 0.02}$	-	$61.69_{\pm 0.09}$	-
Delta	$97.99_{\pm 0.06}$	-	$65.72_{\pm 0.17}$	-	$52.75_{\pm 0.12}$	-	$57.48_{\pm 0.12}$	-
Unigram	$99.83_{\pm 0.10}$	-	$99.63_{\pm 0.19}$	-	$\mathbf{87.79}_{\pm 1.92}$	-	$\underline{94.38_{\pm 1.05}}$	-
KGW	$\mathbf{99.97}_{\pm 0.00}$	-	$99.66_{\pm 0.00}$	-	$81.05_{\pm 0.23}$	-	$92.34_{\pm 0.09}$	-
KGW-Pool	$\underline{99.96_{\pm 0.00}}$	$-0.01_{\pm 0.00}$	$\underline{99.75_{\pm 0.02}}$	$0.09_{\pm 0.02}$	$\underline{87.41_{\pm 0.58}}$	$6.36_{\pm 0.38}$	$94.29_{\pm 0.09}$	$1.95_{\pm 0.12}$
EXP	$99.86_{\pm 0.02}$	-	$99.70_{\pm 0.04}$	-	$80.55_{\pm 0.41}$	-	$91.17_{\pm 0.06}$	-
EXP-Pool	$99.92_{\pm 0.01}$	$0.06_{\pm 0.02}$	$\mathbf{99.83}_{\pm 0.01}$	$0.13_{\pm 0.04}$	$85.87_{\pm 0.32}$	$5.32_{\pm 0.69}$	$\mathbf{96.05}_{\pm 0.05}$	$4.88_{\pm 0.11}$
ITS	$97.96_{\pm 0.06}$	-	$88.41_{\pm 0.19}$	-	$63.12_{\pm 0.10}$	-	$68.02_{\pm 0.21}$	-
ITS-Pool	$99.75_{\pm 0.00}$	$1.79_{\pm 0.06}$	$98.48_{\pm 0.01}$	$10.07_{\pm 0.19}$	$66.35_{\pm 0.13}$	$3.23_{\pm 0.22}$	$81.99_{\pm 0.07}$	$13.97_{\pm 0.27}$

Table 6: Efficacy and Robustness of different watermarking methods on OPT-1.3B evaluated with ROC-AUC. Δ is the performance boost brought by WaterPool. The best and second-best results are highlighted in **bold** and underline.

nario in watermarking as more and more non-watermarked LLMs will emerge. Subsequently, we evaluate the performance of the retrieval watermark across eight groups of watermark detection experiments. The robustness, evaluated by TPR@FPR=1% under lexical attack, is presented in Table 10. The results show that even under lexical attack, the weakest attack, the retrieval watermark experiences significant performance degradation of more than 40%, while most other watermarking techniques maintain high TPR@FPR=1% (see Table 2). This phenomenon substantiate our claims in Section 3.4, highlighting the vulnerability of retrieval watermark compared to other methods in practical applications.

E.3 Performance with Diverse Negative Samples

Full results are presented in Table 11.

E.4 Scaling Length of Watermarked Texts

Both (Kirchenbauer et al., 2023a) and (Kirchenbauer et al., 2023b) observe that the detection rate of watermarking is a monotonically increasing function of the watermarked text length. In this section, we conduct an experiment to investigate the performance of WaterPool with growths of text length. We generate outputs of different lengths $T \in [80, 90, ...200]$ and calculate the corresponding TPR@FPR=1% metrics. The results are presented in Figure 4. We observe a consistent increase in the detection rate of WaterPool under different attack settings, aligning with the findings reported by (Kirchenbauer et al., 2023b). Moreover, WaterPool consistently enhance the performance of original watermarking techniques across all settings, further underscoring its superior capabilities.

	Glob-distinct2		Glob-distinct3		Group-distinct2		Group-distinct3		Perplexity	
	value↑	Δ↑	value↑	Δ↑	value↑	Δ↑	value↑	Δ↑	value↓	Δ↓
Open-Ended Text Generation										
Non-watermark	$39.4_{\pm0.9}$	$0.0_{\pm0.0}$	$75.8_{\pm2.0}$	$0.0_{\pm0.0}$	$84.3_{\pm2.3}$	$0.0_{\pm0.0}$	$94.1_{\pm2.7}$	$0.0_{\pm0.0}$	$\underline{6.8_{\pm0.0}}$	$\underline{0.0_{\pm0.0}}$
Gamma	$39.9_{\pm0.0}$	$0.5_{\pm0.9}$	$77.0_{\pm0.0}$	$1.2_{\pm2.0}$	$85.6_{\pm0.0}$	$1.3_{\pm2.3}$	$95.6_{\pm0.0}$	$1.5_{\pm2.7}$	$6.8_{\pm0.0}$	$0.0_{\pm0.0}$
Delta	$39.9_{\pm0.0}$	$0.5_{\pm0.8}$	$77.0_{\pm0.0}$	$1.2_{\pm2.0}$	$85.6_{\pm0.0}$	$1.3_{\pm2.3}$	$\underline{95.7_{\pm0.0}}$	$\underline{1.6_{\pm2.7}}$	$6.8_{\pm0.0}$	$0.0_{\pm0.0}$
Unigram	$36.5_{\pm1.8}$	$-2.9_{\pm0.9}$	$72.4_{\pm2.4}$	$-3.5_{\pm1.2}$	$82.9_{\pm2.2}$	$-1.4_{\pm1.5}$	$94.8_{\pm0.7}$	$0.7_{\pm2.3}$	$8.8_{\pm0.4}$	$1.9_{\pm0.4}$
KGW	$38.2_{\pm0.0}$	$-1.2_{\pm0.9}$	$74.8_{\pm0.0}$	$-1.0_{\pm2.0}$	$85.3_{\pm0.0}$	$1.0_{\pm2.3}$	$95.6_{\pm0.0}$	$1.5_{\pm2.7}$	$8.4_{\pm0.0}$	$1.6_{\pm0.0}$
KGW-Pool	$\mathbf{41.9}_{\pm0.1}$	$\mathbf{2.5}_{\pm0.8}$	$\mathbf{79.7}_{\pm0.0}$	$\mathbf{3.8}_{\pm2.0}$	$\mathbf{87.4}_{\pm0.0}$	$\mathbf{3.1}_{\pm2.3}$	$\mathbf{96.7}_{\pm0.0}$	$\mathbf{2.6}_{\pm2.7}$	$8.8_{\pm0.0}$	$2.0_{\pm0.0}$
EXP	$32.0_{\pm0.1}$	$-7.4_{\pm0.9}$	$61.9_{\pm0.4}$	$-13.9_{\pm2.3}$	$73.0_{\pm0.3}$	$-11.3_{\pm2.2}$	$81.9_{\pm0.2}$	$-12.2_{\pm2.7}$	$6.9_{\pm0.0}$	$0.0_{\pm0.0}$
EXP-Pool	$\underline{39.9_{\pm0.0}}$	$\underline{0.6_{\pm0.9}}$	$\underline{77.0_{\pm0.0}}$	$\underline{1.2_{\pm2.0}}$	$85.6_{\pm0.0}$	$1.3_{\pm2.3}$	$95.7_{\pm0.0}$	$1.6_{\pm2.7}$	$6.8_{\pm0.0}$	$0.0_{\pm0.0}$
ITS	$35.9_{\pm0.0}$	$-3.5_{\pm0.9}$	$68.2_{\pm0.1}$	$-7.6_{\pm2.0}$	$75.9_{\pm0.1}$	$-8.4_{\pm2.2}$	$84.5_{\pm0.1}$	$-9.6_{\pm2.6}$	$\mathbf{6.6}_{\pm0.0}$	$\mathbf{-0.3}_{\pm0.0}$
ITS-Pool	$39.9_{\pm0.0}$	$0.5_{\pm0.9}$	$77.0_{\pm0.0}$	$1.2_{\pm2.0}$	$\underline{85.6_{\pm0.0}}$	$\underline{1.3_{\pm2.3}}$	$95.6_{\pm0.0}$	$1.6_{\pm2.7}$	$6.8_{\pm0.0}$	$0.0_{\pm0.0}$
Long-Form Question Answering										
Non-watermark	$\underline{33.0_{\pm0.0}}$	$\underline{0.0_{\pm0.0}}$	$\underline{71.3_{\pm0.0}}$	$\underline{0.0_{\pm0.0}}$	$\mathbf{87.1}_{\pm0.1}$	$\mathbf{0.0}_{\pm0.0}$	$\mathbf{97.1}_{\pm0.0}$	$\mathbf{0.0}_{\pm0.0}$	$8.8_{\pm0.0}$	$0.0_{\pm0.0}$
Gamma	$32.9_{\pm0.0}$	$-0.1_{\pm0.0}$	$71.2_{\pm0.1}$	$-0.1_{\pm0.1}$	$87.1_{\pm0.0}$	$-0.1_{\pm0.0}$	$97.0_{\pm0.0}$	$-0.1_{\pm0.0}$	$8.8_{\pm0.0}$	$-0.0_{\pm0.0}$
Delta	$33.0_{\pm0.0}$	$-0.0_{\pm0.0}$	$71.2_{\pm0.1}$	$-0.1_{\pm0.1}$	$87.1_{\pm0.0}$	$-0.1_{\pm0.1}$	$97.0_{\pm0.0}$	$-0.0_{\pm0.0}$	$8.8_{\pm0.0}$	$-0.0_{\pm0.1}$
Unigram	$29.2_{\pm2.0}$	$-3.8_{\pm2.0}$	$65.1_{\pm2.8}$	$-6.2_{\pm2.8}$	$82.5_{\pm2.2}$	$-4.7_{\pm2.2}$	$95.1_{\pm0.6}$	$-2.0_{\pm0.6}$	$10.4_{\pm0.8}$	$1.6_{\pm0.8}$
KGW	$31.3_{\pm0.1}$	$-1.7_{\pm0.1}$	$68.1_{\pm0.1}$	$-3.2_{\pm0.1}$	$86.0_{\pm0.0}$	$-1.1_{\pm0.0}$	$96.5_{\pm0.0}$	$-0.6_{\pm0.0}$	$10.8_{\pm0.0}$	$2.0_{\pm0.0}$
KGW-Pool	$\mathbf{34.8}_{\pm0.0}$	$\mathbf{1.8}_{\pm0.0}$	$\mathbf{73.5}_{\pm0.0}$	$\mathbf{2.2}_{\pm0.1}$	$85.2_{\pm0.1}$	$-2.0_{\pm0.1}$	$95.1_{\pm0.0}$	$-2.0_{\pm0.0}$	$10.5_{\pm0.0}$	$1.6_{\pm0.0}$
EXP	$25.2_{\pm0.8}$	$-7.8_{\pm0.8}$	$54.1_{\pm1.8}$	$-17.2_{\pm1.8}$	$75.3_{\pm0.1}$	$-11.8_{\pm0.1}$	$84.7_{\pm0.1}$	$-12.4_{\pm0.1}$	$\underline{8.7_{\pm0.2}}$	$\underline{-0.1_{\pm0.2}}$
EXP-Pool	$32.9_{\pm0.1}$	$-0.1_{\pm0.1}$	$71.1_{\pm0.3}$	$-0.1_{\pm0.2}$	$\underline{87.1_{\pm0.0}}$	$\underline{-0.1_{\pm0.0}}$	$\underline{97.1_{\pm0.0}}$	$\underline{-0.0_{\pm0.0}}$	$8.8_{\pm0.0}$	$-0.0_{\pm0.0}$
ITS	$29.4_{\pm0.0}$	$-3.6_{\pm0.1}$	$62.7_{\pm0.1}$	$-8.6_{\pm0.1}$	$77.4_{\pm0.1}$	$-9.8_{\pm0.1}$	$85.9_{\pm0.1}$	$-11.2_{\pm0.1}$	$\mathbf{8.4}_{\pm0.0}$	$\mathbf{-0.5}_{\pm0.0}$
ITS-Pool	$32.9_{\pm0.0}$	$-0.1_{\pm0.0}$	$71.1_{\pm0.0}$	$-0.1_{\pm0.0}$	$87.1_{\pm0.0}$	$-0.1_{\pm0.0}$	$97.0_{\pm0.0}$	$-0.1_{\pm0.0}$	$8.8_{\pm0.0}$	$-0.0_{\pm0.0}$

Table 7: Imperceptibility of different watermarking methods on OPT-6.7B. Δ is the difference between watermarked and non-watermarked texts. The best and second-best results before rounding are highlighted in **bold** and $\underline{\text{underline}}$.

	w/o Attack		Lexical-Attack		Dipper-Attack		Translation-Attack	
	value↑	Δ	value↑	Δ	value↑	Δ	value↑	Δ
Open-Ended Text Generation								
Gamma	$95.46_{\pm0.07}$	-	$16.08_{\pm0.44}$	-	$2.40_{\pm0.12}$	-	$2.93_{\pm0.09}$	-
Delta	$70.85_{\pm0.11}$	-	$7.34_{\pm0.26}$	-	$2.10_{\pm0.04}$	-	$2.60_{\pm0.05}$	-
Unigram	$93.68_{\pm2.32}$	-	$89.39_{\pm4.31}$	-	$\underline{23.62_{\pm13.20}}$	-	$36.52_{\pm14.15}$	-
KGW	$\mathbf{97.58}_{\pm0.08}$	-	$86.17_{\pm0.49}$	-	$14.63_{\pm0.05}$	-	$25.88_{\pm0.16}$	-
KGW-Pool	$\underline{96.77_{\pm0.11}}$	$-0.81_{\pm0.03}$	$\underline{93.30_{\pm0.08}}$	$7.13_{\pm0.42}$	$\mathbf{23.66}_{\pm1.03}$	$9.03_{\pm1.06}$	$\underline{37.91_{\pm0.12}}$	$12.03_{\pm0.10}$
EXP	$94.84_{\pm0.35}$	-	$88.97_{\pm0.68}$	-	$15.40_{\pm1.24}$	-	$27.16_{\pm1.28}$	-
EXP-Pool	$96.56_{\pm0.96}$	$1.72_{\pm1.31}$	$\mathbf{94.19}_{\pm0.07}$	$5.22_{\pm0.66}$	$22.03_{\pm0.85}$	$6.63_{\pm2.05}$	$\mathbf{43.84}_{\pm0.52}$	$16.68_{\pm0.89}$
ITS	$64.71_{\pm0.46}$	-	$21.06_{\pm0.59}$	-	$2.14_{\pm0.08}$	-	$2.95_{\pm0.17}$	-
ITS-Pool	$88.43_{\pm0.11}$	$23.72_{\pm0.47}$	$60.89_{\pm0.34}$	$39.84_{\pm0.57}$	$3.77_{\pm0.06}$	$1.63_{\pm0.13}$	$8.98_{\pm0.11}$	$6.03_{\pm0.20}$
Long-Form Question Answering								
Gamma	$98.33_{\pm0.04}$	-	$20.20_{\pm0.39}$	-	$2.22_{\pm0.05}$	-	$4.34_{\pm0.12}$	-
Delta	$89.08_{\pm0.16}$	-	$12.08_{\pm0.10}$	-	$2.11_{\pm0.13}$	-	$4.22_{\pm0.15}$	-
Unigram	$96.50_{\pm2.59}$	-	$90.67_{\pm6.19}$	-	$\underline{29.91_{\pm13.55}}$	-	$42.00_{\pm19.57}$	-
KGW	$\underline{99.29_{\pm0.01}}$	-	$92.69_{\pm0.30}$	-	$17.59_{\pm0.26}$	-	$41.33_{\pm0.24}$	-
KGW-Pool	$\mathbf{99.40}_{\pm0.17}$	$0.11_{\pm0.17}$	$\underline{97.25_{\pm0.17}}$	$4.56_{\pm0.23}$	$28.17_{\pm0.84}$	$10.57_{\pm0.83}$	$45.67_{\pm2.06}$	$4.33_{\pm1.88}$
EXP	$98.66_{\pm0.08}$	-	$96.11_{\pm0.21}$	-	$22.41_{\pm1.82}$	-	$\underline{49.18_{\pm0.97}}$	-
EXP-Pool	$99.27_{\pm0.04}$	$0.61_{\pm0.06}$	$\mathbf{98.12}_{\pm0.03}$	$2.01_{\pm0.23}$	$\mathbf{32.14}_{\pm0.08}$	$9.73_{\pm1.89}$	$\mathbf{65.61}_{\pm0.06}$	$16.43_{\pm0.92}$
ITS	$81.56_{\pm0.37}$	-	$33.43_{\pm0.74}$	-	$2.77_{\pm0.07}$	-	$6.38_{\pm0.19}$	-
ITS-Pool	$96.48_{\pm0.08}$	$14.92_{\pm0.37}$	$78.37_{\pm0.36}$	$44.94_{\pm0.66}$	$6.10_{\pm0.26}$	$3.33_{\pm0.21}$	$20.88_{\pm0.36}$	$14.50_{\pm0.44}$

Table 8: Efficacy and Robustness of different watermarking methods on OPT-6.7B evaluated with TPR@FPR=1%. Δ is the performance boost brought by WaterPool. The best and second-best results before rounding are highlighted in **bold** and $\underline{\text{underline}}$.

	w/o Attack		Lexical-Attack		Dipper-Attack		Translation-Attack	
	value↑	Δ	value↑	Δ	value↑	Δ	value↑	Δ
Open Text Generation								
Gamma	99.33$_{\pm 0.01}$	-	77.01$_{\pm 0.14}$	-	55.20$_{\pm 0.17}$	-	57.47$_{\pm 0.20}$	-
Delta	92.84$_{\pm 0.05}$	-	61.06$_{\pm 0.26}$	-	52.39$_{\pm 0.07}$	-	53.83$_{\pm 0.04}$	-
Unigram	99.48$_{\pm 0.19}$	-	99.12$_{\pm 0.38}$	-	**83.94**$_{\pm 5.55}$	-	$\underline{89.73_{\pm 3.35}}$	-
KGW	$\underline{99.77_{\pm 0.01}}$	-	98.96$_{\pm 0.03}$	-	76.73$_{\pm 0.17}$	-	83.73$_{\pm 0.16}$	-
KGW-Pool	**99.84**$_{\pm 0.00}$	0.06$_{\pm 0.01}$	**99.62**$_{\pm 0.00}$	0.66$_{\pm 0.03}$	$\underline{83.88_{\pm 0.71}}$	7.14$_{\pm 0.76}$	**90.85**$_{\pm 0.09}$	7.13$_{\pm 0.23}$
EXP	99.12$_{\pm 0.03}$	-	98.32$_{\pm 0.04}$	-	71.72$_{\pm 0.49}$	-	78.93$_{\pm 1.57}$	-
EXP-Pool	99.46$_{\pm 0.21}$	0.34$_{\pm 0.18}$	$\underline{99.29_{\pm 0.03}}$	0.96$_{\pm 0.07}$	77.78$_{\pm 1.12}$	6.06$_{\pm 1.61}$	88.22$_{\pm 0.13}$	9.29$_{\pm 1.45}$
ITS	93.25$_{\pm 0.06}$	-	79.89$_{\pm 0.10}$	-	59.15$_{\pm 0.18}$	-	55.34$_{\pm 0.19}$	-
ITS-Pool	98.63$_{\pm 0.01}$	5.38$_{\pm 0.07}$	95.40$_{\pm 0.06}$	15.51$_{\pm 0.07}$	61.17$_{\pm 0.12}$	2.02$_{\pm 0.23}$	69.49$_{\pm 0.10}$	14.15$_{\pm 0.16}$
Long-Form Question Answering								
Gamma	99.81$_{\pm 0.00}$	-	80.10$_{\pm 0.11}$	-	54.91$_{\pm 0.06}$	-	60.16$_{\pm 0.05}$	-
Delta	97.74$_{\pm 0.02}$	-	65.49$_{\pm 0.07}$	-	52.54$_{\pm 0.12}$	-	56.29$_{\pm 0.07}$	-
Unigram	99.80$_{\pm 0.14}$	-	99.57$_{\pm 0.26}$	-	**88.43**$_{\pm 4.17}$	-	$\underline{93.74_{\pm 2.41}}$	-
KGW	**99.96**$_{\pm 0.00}$	-	99.57$_{\pm 0.00}$	-	80.23$_{\pm 0.16}$	-	91.00$_{\pm 0.08}$	-
KGW-Pool	$\underline{99.95_{\pm 0.01}}$	-0.01$_{\pm 0.01}$	$\underline{99.73_{\pm 0.01}}$	0.17$_{\pm 0.00}$	$\underline{86.84_{\pm 0.62}}$	6.61$_{\pm 0.49}$	93.32$_{\pm 0.04}$	2.31$_{\pm 0.10}$
EXP	99.81$_{\pm 0.01}$	-	99.54$_{\pm 0.03}$	-	77.31$_{\pm 1.28}$	-	89.34$_{\pm 0.34}$	-
EXP-Pool	99.90$_{\pm 0.00}$	0.09$_{\pm 0.01}$	**99.77**$_{\pm 0.01}$	0.23$_{\pm 0.02}$	84.38$_{\pm 0.15}$	7.07$_{\pm 1.16}$	**94.71**$_{\pm 0.02}$	5.36$_{\pm 0.34}$
ITS	97.04$_{\pm 0.02}$	-	86.25$_{\pm 0.04}$	-	62.49$_{\pm 0.14}$	-	65.94$_{\pm 0.10}$	-
ITS-Pool	99.67$_{\pm 0.01}$	2.63$_{\pm 0.01}$	98.08$_{\pm 0.01}$	11.83$_{\pm 0.04}$	65.85$_{\pm 0.13}$	3.36$_{\pm 0.28}$	80.03$_{\pm 0.10}$	14.10$_{\pm 0.18}$

Table 9: Efficacy and Robustness of different watermarking methods on OPT-6.7B evaluated with ROC-AUC. Δ is the performance boost brought by WaterPool. The best and second-best results are highlighted in **bold** and $\underline{\text{underline}}$.

Watermarked Model	Vicuna-13b	Vicuna-7b	Llama2-13b	Llama2-7b	Gemma-2b	Gemma-7b	OPT-1.3b	OPT-6.7b	Avg
Open-Ended Text Generation									
Vicuna-13b	-	16.36	32.36	40.85	61.23	51.09	49.94	40.02	41.69
Vicuna-7b	14.62	-	39.24	41.41	61.16	53.00	49.63	39.73	42.69
Llama2-13b	23.36	31.65	-	44.56	66.95	58.67	53.12	43.22	45.93
Llama2-7b	29.19	31.63	41.54	-	68.24	60.53	53.10	43.35	46.80
Gemma-2b	47.32	48.20	63.43	65.86	-	67.78	63.05	54.13	58.54
Gemma-7b	37.95	41.03	55.49	58.77	68.70	-	58.30	48.65	52.70
OPT-1.3b	42.29	42.29	53.00	57.93	66.59	62.43	-	53.00	53.93
OPT-6.7b	30.73	30.73	47.83	47.83	62.37	57.79	52.99	-	47.18
Long-Form Question Answering									
Vicuna-13b	-	0.87	39.39	44.02	68.47	45.95	55.36	46.01	42.87
Vicuna-7b	0.76	-	40.00	42.84	68.19	45.65	54.25	45.01	42.38
Llama2-13b	9.16	9.81	-	44.00	66.52	46.48	50.77	41.62	38.34
Llama2-7b	10.77	10.37	41.94	-	66.40	47.64	53.99	45.15	39.47
Gemma-2b	21.65	21.88	64.71	67.15	-	65.23	69.16	60.29	52.87
Gemma-7b	8.93	9.56	42.78	45.74	63.75	-	54.14	44.80	38.53
OPT-1.3b	17.56	17.56	47.64	47.64	64.59	52.26	-	52.26	42.79
OPT-6.7b	12.83	8.93	41.87	41.87	59.81	41.87	55.68	-	37.55

Table 10: TPR@FPR=1% of retrieval watermark under lexical attacks. The model in the first column is the model being watermarked. Retrieval watermark is vulnerable even under the weakest lexical attacks.

		w/o Attack	Lexical-Attack	Dipper-Attack	Translation-Attack
EXP-Pool	Original	98.43 / 99.56	96.67 / 98.81	26.17 / 36.24	51.41 / 72.61
	Human	98.35 / 99.66	96.50 / 99.04	24.49 / 39.70	50.23 / 75.00
	Other Models	98.45 / 99.61	96.72 / 98.88	25.65 / 36.63	51.60 / 72.51
KGW-Pool	Original	98.29 / 99.51	95.29 / 97.97	24.62 / 29.92	42.26 / 50.14
	Human	98.96 / 99.98	96.27 / 99.56	27.23 / 46.80	46.22 / 69.24
	Other Models	98.96 / 99.71	96.21 / 98.03	26.33 / 30.52	45.42 / 51.66
ITS-Pool	Original	92.56 / 97.56	68.50 / 81.73	4.05 / 6.25	10.83 / 24.26
	Human	93.33 / 97.44	71.18 / 80.77	4.61 / 5.82	12.01 / 22.95
	Other Models	92.73 / 97.65	69.27 / 82.22	4.15 / 6.62	11.17 / 24.92

Table 11: TPR@FPR=1% of WaterPool with different non-watermarked texts. The results are in form of `(C4 Result / LFQA Result)`. The first column lists watermarking methods, and the second column shows non-watermarked text sources. WaterPool remains stable across different non-watermarked texts.

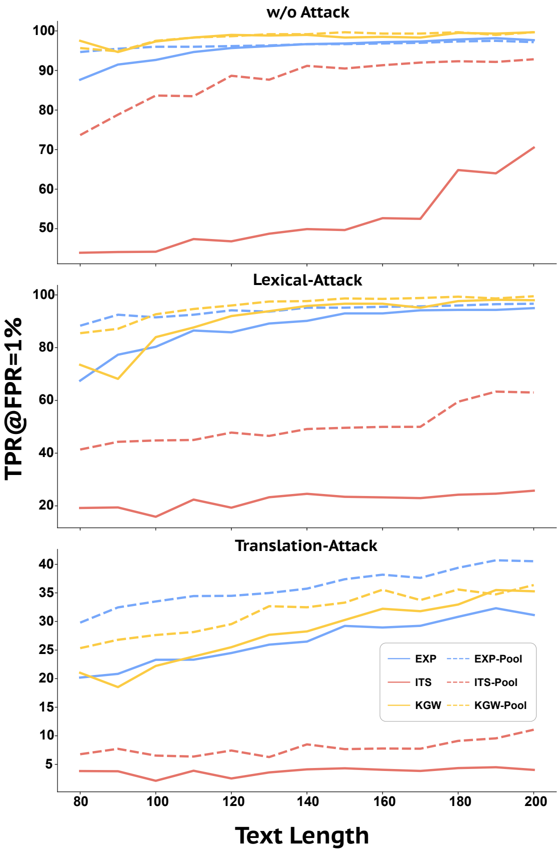

Figure 4: TPR@FPR=1% of different watermarking techniques with the growths of text length. The same color indicates different methods sharing the same mark module. Solid lines represent original methods while dashed lines represent WaterPool methods.