Sequential Watermarking of Language Models [1][2][3]

1 Abstract

With the rise of large language models (LLMs), the ability to verify the provenance of any piece of text has become increasingly critical. A widely accepted solution is that of watermarking techniques. Many schemes based on cryptography, syntax, and statistics, among others have emerged. While these methods differ in the mechanism used to embed the watermark, a common aspect across approaches is the statistical inference procedure employed during watermark detection. In this report, we investigate the problem of sequentializing the detection process. We demonstrate that sequential detection enables the possibility of early stopping, thereby reducing computational overhead without compromising the robustness of the watermarking scheme. To assess the effectiveness of our methods, we apply them to a language models—OPT-1.3B and experimentally evaluate their statistical power and robustness against various paraphrasing attacks. Our results show that for OPT-1.3B model, watermarked text can be reliably detected (with $p \leq 0.05$) using as few as 40 permutations, even after introducing corruption by randomly editing 40–50% of the tokens through substitutions, insertions, or deletions.

2 Introduction

Traditional watermarking methods, which leveraged clear distinctions between machine-generated and human-written content, are becoming increasingly ineffective as LLM capabilities advance [9, 3]. In parallel, more sophisticated detection strategies have emerged [1, 5, 6]. Recent efforts have focused on statistical embedding and detection mechanisms that satisfy three key desiderata for a practical watermark: model- and prompt-agnosticity, distortion-free text generation, and robustness against adversarial attacks. Kuditipudi [8] introduced a distortion-free watermarking scheme with strong empirical robustness; however, its reliance on a permutation test limits its scalability and online applicability. In this project, we address this bottleneck by proposing a sequential detection algorithm [2] based on an anytime-valid p-process, inspired by the work of Fischer and Ramdas [4]. This sequentialization enables faster, online watermark detection while retaining strong theoretical guarantees under the testing-by-betting framework. Furthermore, our approach naturally supports early stopping, futility analysis, on-the-fly p-value access, and reduced computational overhead.

Recent advances in LLM watermarking trace back to the Red-Green List algorithm by Kirchenbauer et al. [7], which partitions the vocabulary into red and green sets and biases sampling toward green tokens. Detection is based on counting "violations"—uses of red tokens—with a high count suggesting human authorship. Although effective, this method introduces syntactic distortion and requires white-box access to model logits.

To address these issues, Aaronson et al. [9] proposed a watermark based on accumulating evidence through boosted token parameters. While removing the need for white-box access, their method remains vulnerable to adversarial substitution attacks that can erase the watermark.

Kuditipudi et al. [8] further advanced the field by introducing a distortion-free and robust watermarking technique. Their approach modifies the token sampling process using a pseudo-random number generator (PRNG) keyed by a shared secret ξ, ensuring that, marginalizing over the randomness of ξ, the output distribution remains identical to that of the original model.

[1] All experimental codes and results are available on Github

[2] Contributions have been included at the end with the references

[3] Relevant background and mathematical definitions have been added as appendices

Formally, the distribution

$$P(\text{text}) = \int_{\xi} \mathbf{1}\{\text{text} = \text{generate}(\xi, \text{prompt})\}\, d\nu(\xi)$$

is equivalent to the native language model distribution. This distortion-free property guarantees that watermarked and non-watermarked outputs are indistinguishable without knowledge of the key. This framework enables the computation of p-values under the null hypothesis that the text is independent of the watermark key sequence. For detection, the observed text is aligned to the key sequence ξ using a robust sequence alignment algorithm. The alignment cost $\phi(\text{text}, \xi)$ is used as a test statistic. To compute a p-value, the detector generates many random keys $\xi^{(t)}$ and compares the alignment cost with the true key:

$$p = \frac{1 + \sum_{t=1}^{T} \mathbb{I}[\phi(\text{text}, \xi^{(t)}) \leq \phi(\text{text}, \xi)]}{T + 1}.$$

This forms a permutation test based on alignment costs, ensuring false positive (Type I) error control.

In this project, we extend the methodology introduced by Kuditipudi [8], with a particular focus on enhancing the detection algorithm. Specifically, we substitute the original permutation-based hypothesis testing procedure with a sequential testing framework based on the testing-by-betting paradigm, which leverages martingale constructions for statistical inference. The theoretical underpinnings of this sequential framework are discussed in detail later in the report.

To assess the validity of our proposed hypothesis test—which tests for independence between the shared key sequence and an observed text sample—we empirically evaluate its performance by measuring the statistical power and controlling the type-I error rate across varied numbers of permutations. Our experimental results demonstrate that the sequential test typically stops after observing approximately the same number of samples required for the permutation test to achieve near-perfect power and a controlled false-positive rate, thus highlighting the adaptive efficiency of our method.

We begin by reviewing the proposed method in Section 3. In Section 4, we introduce our theoretical guarantees, followed by a discussion of the experimental setup in Section 5. Results are presented in Section 6. Finally, we conclude the report in Section 7. Furthermore, we provide all necessary mathematical definitions in the Appendix.

3 Proposed Method

We consider testing the null hypothesis H_0 that a given text Y is not watermarked (i.e., independent of the watermark key) against the alternative H_1 that it is watermarked. Let $\phi(Y, \xi)$ be the detector's test statistic (e.g., an alignment cost between Y and key ξ), where lower values of ϕ indicate stronger evidence of a watermark.

Denote the observed test statistic by $S_0 = \phi(Y, \xi_0)$, where ξ_0 is the true key, and let $S_i = \phi(Y, \xi_i)$ be the statistic computed with a random key ξ_i sampled under H_0. Under H_0, the values $\{S_0, S_1, S_2, \dots\}$ are exchangeable.

Rather than performing a fixed-sample permutation test, we propose a sequential hypothesis test based on a nonnegative martingale process W_n with $W_0 = 1$, constructed as follows.

1. Test Statistic and Betting Process: At iteration n, we compute the rank R_n of S_0 among $\{S_0, S_1, \dots, S_n\}$. We then select a betting factor $\lambda_n(r)$ satisfying

$$\frac{1}{n+1} \sum_{r=0}^{n} \lambda_n(r) = 1,$$

and update the wealth process by

$$W_n = W_{n-1}\lambda_n(R_n).$$

The sequence $\{W_n\}$ remains a nonnegative martingale under H_0.

Sequential Decision Rule: The test operates with two thresholds $0 < B < 1 < A$.

- If $W_n \geq A$, we stop and reject H_0 (declare watermark present).
- If $W_n \leq B$, we stop and accept H_0 (declare watermark absent).
- Otherwise, we continue sampling.

To guarantee Type I error control at level α, we set $A = \frac{1}{\alpha}$. The final p-value at time n can be reported as $p_n = \frac{1}{\sup_{k \leq n} W_k}$.

Unlike fixed-sample permutation tests, this sequential method continuously monitors evidence and allows early stopping without pre-specifying the sample size. The permutation test produces a single p-value after all samples are drawn, while the sequential test maintains an anytime-valid p-value p_n at each time n.

4 Theoretical Analysis

Error Control Guarantee: Under the null hypothesis H_0, the wealth process $\{W_n\}$ is a nonnegative martingale with initial value $W_0 = 1$. By Ville's inequality,

$$\mathbb{P}_{H_0}\left(\sup_{n \geq 0} W_n \geq \frac{1}{\alpha}\right) \leq \alpha,$$

which ensures that the sequential test controls the Type I error at level α.

Thus, stopping when $W_n \geq A$ with $A = 1/\alpha$ guarantees that

$$\mathbb{P}_{H_0}(\text{Reject } H_0) \leq \alpha.$$

Stopping Time and Expected Sample Size: Under the alternative hypothesis H_1 (watermark present), the log-wealth process

$$L_n = \log W_n = \sum_{i=1}^{n} Z_i$$

has i.i.d. increments Z_i with ${}_{H_1}[Z_i] = \delta = D_{KL}(P_1 \,\|\, P_0) > 0$. By Wald's identity [10, Thm. 2.2.1],

$$\big[L_\tau\big] = \Big[\sum_{i=1}^{\tau} Z_i\Big] = [\tau]\,\delta,$$

and a bounded-overshoot argument gives $L_\tau \leq \log A + O(1)$. Combining these shows

$$[\tau] \;=\; \frac{[L_\tau]}{\delta} \;=\; O\Big(\frac{\log A}{\delta}\Big) \;=\; O\Big(\frac{\log(1/\alpha)}{D_{KL}(P_1 \,\|\, P_0)}\Big),$$

where $A = 1/\alpha$.

Thus, sequential testing often achieves faster detection compared to a fixed-size permutation test requiring many samples. The sequential method offers several advantages; It does not require a fixed number of samples upfront, it allows early stopping based on accumulated evidence, and it maintains anytime-valid error guarantees via martingale theory (Ville's inequality).

In contrast, permutation tests require drawing a large number of permutations and computing a final p-value after all data is collected.

Betting Strategy and Martingale Properties: A properly chosen betting strategy (e.g., mixture or Kelly-optimal) ensures that the wealth process is a nonnegative martingale under H_0. This is critical for maintaining Type I error control.

For example, a Kelly-optimal betting strategy aims to maximize the expected log-wealth growth when the alternative holds, enhancing the power of the sequential test.

In summary, the sequential detection method provides valid hypothesis testing with strong theoretical guarantees under H_0, potential computational savings under H_1, and greater flexibility compared to traditional fixed-sample permutation tests. Theoretical analysis based on martingale properties underpins its correctness and practical effectiveness.

An ideal watermark satisfies three key characteristics: it is agnostic, distortion-free, and robust. The watermarking algorithm proposed by Kuditipudi[8], is specifically designed to achieve all three of these attributes. Mathematically, we break down the process into two components: a *generate* method that deterministically maps a sequence of random keys, denoted as ξ, encoded by the watermark key to a sample from the language model, and a detect method that aligns a given watermarked text with the watermark key sequence using the shared key.

To simplify our discussion, we break our work into four algorithms as follows: Watermarked Text Generation, Watermarked Text Detection, Sequential Monte Carlo Test, and Test Statistic.

5 Experimental Setup

The experiments are conducted using Facebook's OPT-1.3B language model, a 1.3-billion parameter open-source model developed by Meta, evaluated on the C4 dataset—a large-scale English-language corpus tailored for language modeling tasks. We test multiple watermarking methods including inverse transform sampling and exponential minimum sampling (both with edit variants), as well as Kirchenbauer watermarking with $\delta = 1$. For each method, we compute p-values and the number of permutations used under both fixed and sequential test frameworks. This comprehensive setup allows us to validate our method across a variety of watermarking strategies and datasets, strengthening the robustness of our findings.

5.1 Oracle Power Curve Experiment

Our experimental setup is centered around an oracle experiment designed to benchmark the performance of our proposed sequential permutation test against the traditional fixed-sample permutation test. The goal is to identify the minimum number of resamples—referred to as the "oracle number"—required by the standard permutation test to achieve full statistical power (power = 1) when detecting watermarked text. To this end, we simulate a large number of trials under the alternative hypothesis using synthetically generated watermarked text. For each trial, we run the traditional permutation test with multiple fixed values of T (the number of permutations) and record whether the null hypothesis is correctly rejected (i.e., p-value $< \alpha$). This allows us to empirically determine the smallest value of T for which the test consistently identifies the watermark, thus establishing our oracle benchmark.

Following the oracle determination, we evaluate our sequential permutation test against this benchmark. The key advantage of our method is its adaptivity: the test accumulates permutations incrementally and terminates early once the p-value falls below the significance threshold α. For each trial, we record the number of permutations used before stopping and compare the distribution of these stopping times with the oracle number. This comparison enables us to assess the efficiency gains of the sequential approach in terms of computational cost while ensuring it

does not compromise detection power. The primary metrics analyzed are empirical power, average number of permutations used, and early stopping frequency.

5.2 Robustness to Text Corruption

To evaluate the robustness of the watermark detection methods, we simulate corruptions in the generated outputs and assess the impact on statistical reliability and computational cost. Specifically, we generate $T = 200$ outputs, each with a fixed text length of $m = 80$ tokens. These outputs are then subjected to corruption through random token-level modifications — insertions, deletions, and substitutions — at varying corruption rates ranging from 0.05 to 0.8.

For each corruption level, we run the detection tests and report four key metrics: the average of median p-values across trials, the null rejection rate (false positive rate), empirical power (true positive rate), and the median number of permutations required to make a decision (with a maximum cap of 5,000 permutations). This setup allows us to observe how resilient each detection method is to varying degrees of textual noise, while also evaluating the efficiency of the algorithm under corrupted conditions.

By analyzing these robustness results, we can determine whether the sequential permutation test maintains its advantages — both in terms of statistical validity and computational efficiency — even when the input text is partially corrupted. This provides a realistic evaluation of watermark detection performance in noisy or adversarial environments.

6 Results

6.1 Evaluation of Watermark Detection Methods

Figure 1 presents the empirical power (left) and null rejection rate (right) across multiple runs of the C4 experiment without corruption. The power plot indicates that both the traditional permutation test and the sequential Monte Carlo (SMC) test converge towards a power close to 1 as the number of runs increases. However, the permutation test achieves high power almost immediately, whereas the sequential method takes more trials to reach the same level.

This early convergence of the permutation test, however, is misleading. The null rejection rate plot shows that the permutation test exhibits inflated false positive rates during the early runs. This over-rejection of the null hypothesis undermines the early power values observed in the left plot, indicating that the permutation test is rejecting the null hypothesis even when it should not. In contrast, the sequential Monte Carlo test demonstrates better error control. Its null rejection rate stays below or near the target significance level ($\alpha = 0.05$), suggesting that it more accurately accepts the null hypothesis when the text is not watermarked.

Therefore, although the permutation test superficially appears to reach high power faster, it does so at the cost of validity. The sequential test, despite taking longer to attain high power, produces more reliable and statistically valid results. This demonstrates the strength of the sequential approach in maintaining rigorous type I error control while eventually achieving comparable detection capability.

In addition to its superior statistical validity, the sequential Monte Carlo (SMC) method also exhibits significant computational efficiency advantages, as illustrated in Figure 2. This figure plots the median number of permutations used under both the null and alternative hypotheses, as a function of the maximum allowed number of permutations.

The blue line representing the traditional (non-sequential) permutation test shows a near-linear growth in the number of permutations used, reaching the maximum allowed limit of 1000 in nearly

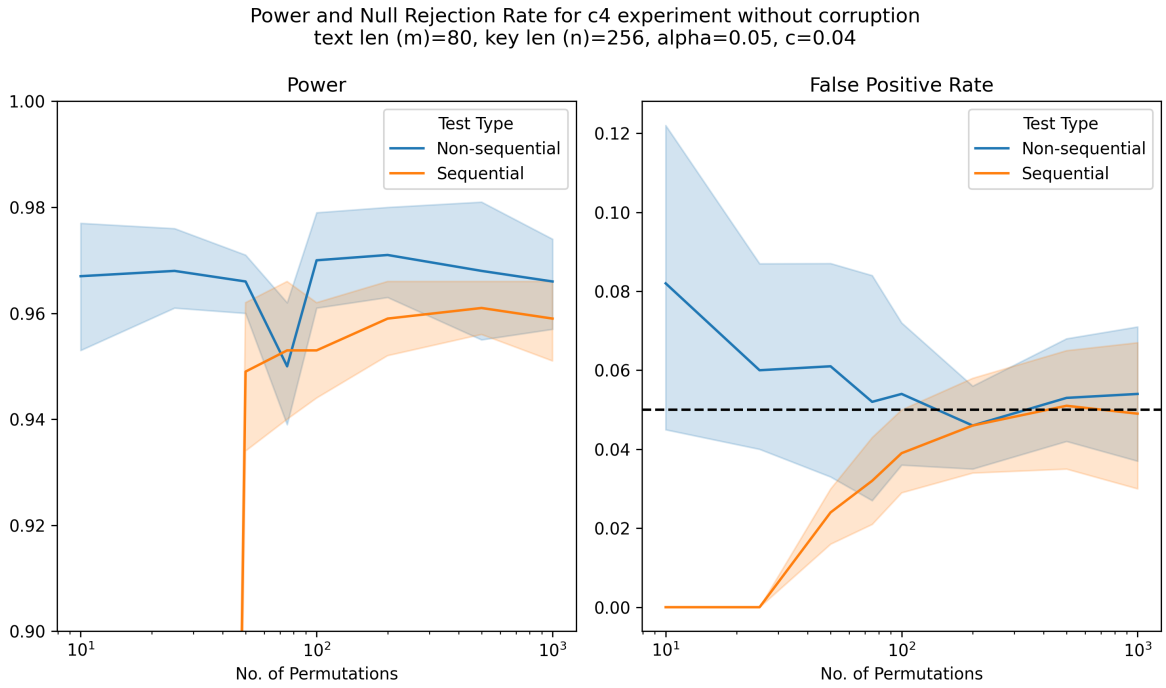


Figure 1: Power and null rejection rate for C4 experiment without corruption. Text length = 80, Key length = 256, Number of texts = 200, $\alpha = 0.05$, $c = 0.04$

all cases. In contrast, the dashed orange line for the SMC method initially grows in tandem with the non-sequential method but quickly plateaus around 50 permutations. This indicates that the SMC method often collects sufficient evidence to reject the null hypothesis far earlier, without exhausting the full permutation budget.

Furthermore, the solid orange line corresponding to the number of permutations used under the null hypothesis shows an early termination of the SMC process. This demonstrates that the SMC test efficiently halts when there is insufficient evidence to reject the null hypothesis, avoiding unnecessary computation. The drop in permutation usage under the null reflects the adaptivity of SMC, not only preserving statistical validity but doing so with dramatically reduced computational cost.

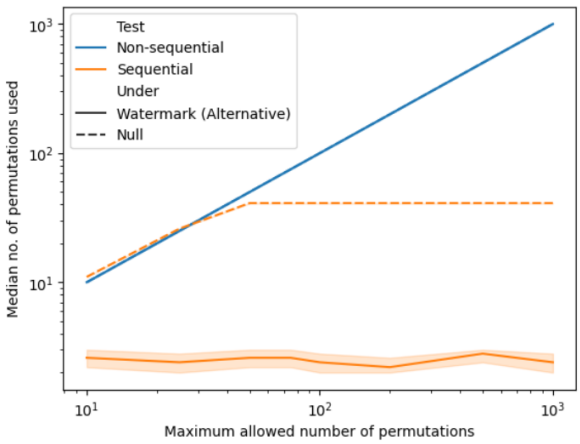
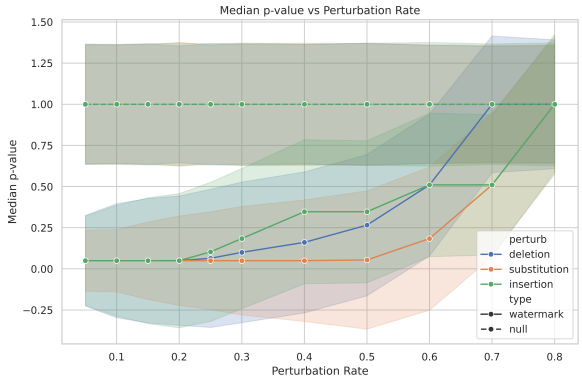


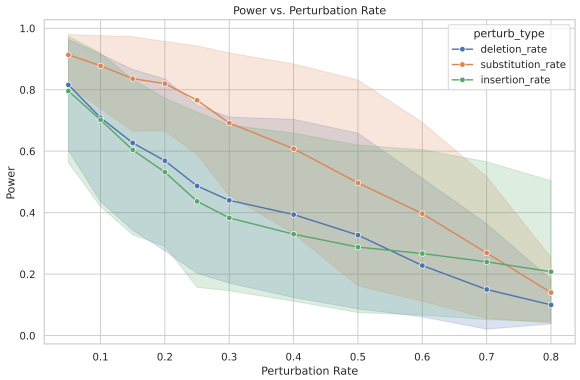
Figure 2: Median number of permutations used by each method under the null and alternative hypotheses. SMC stops early when sufficient evidence is gathered or when the test becomes futile.

6.2 Performance under different corruptions

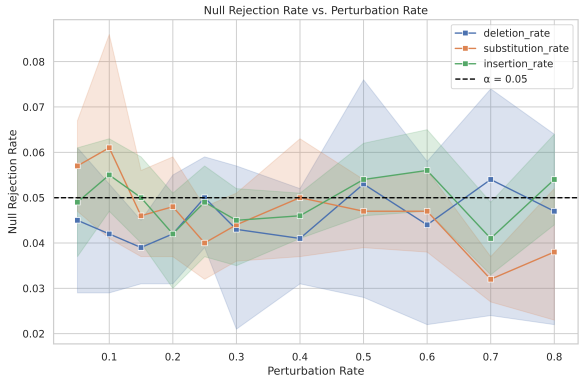
Figure 3 presents the detection results under different types of perturbation attacks. In evaluating the effectiveness of using the Sequential Monte Carlo (SMC) method for watermark detection in models, we draw several key conclusions. First, results of median p-value and empirical power shows that when the perturbation rate is below 0.2, the detection remains highly effective across all types of attacks, demonstrating strong robustness in low-distortion scenarios. Notably, substitution attacks exhibit higher empirical power than deletion and insertion at the same perturbation rates, indicating that such perturbations are easier to detect. This is further supported by consistently lower median p-values for substitution, especially in the 0.2–0.5 range, highlighting the increased sensitivity of our method to this type of modification. Second, the null rejection rate remains consistently close to the significance level ($\alpha = 0.05$) across all perturbation rates, demonstrating good control of false positives. Finally, in terms of computational efficiency, the SMC test significantly reduces the number of permutations required per detection—averaging fewer than 40 compared to the fixed 5000 permutations used in standard permutation tests—highlighting its high efficiency.



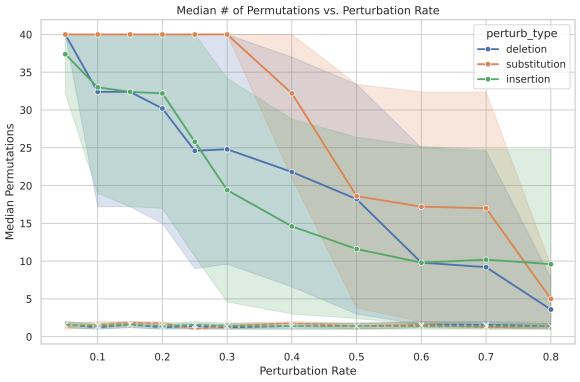
(a) Median p-value



(b) Empirical Power



(c) Null Rejection Rate



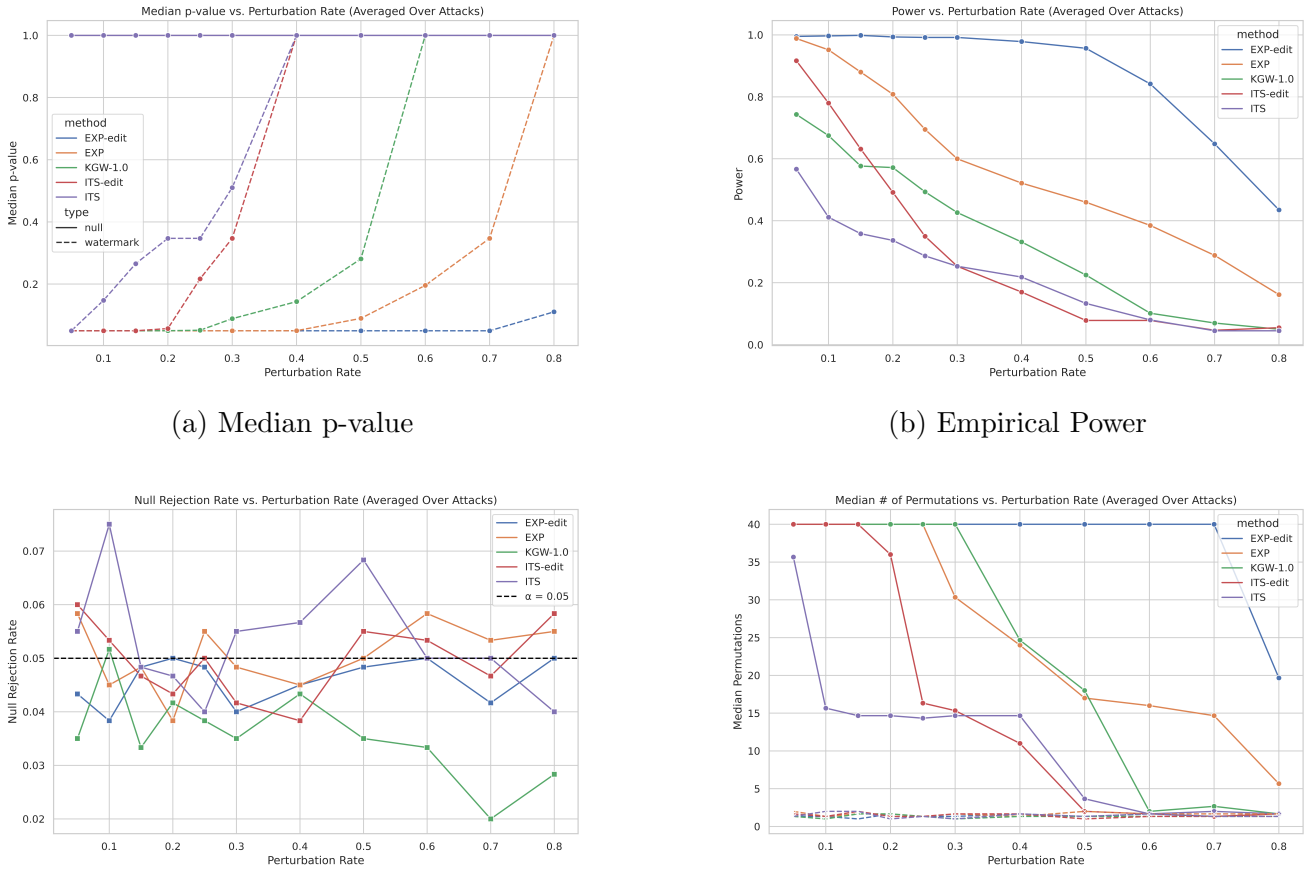
(d) Number of Permutation

Figure 3: Performance Evaluation under Different Perturbation Types

6.3 Robustness Performance of Watermarking Methods

Besides comparing different types of corruptions, we also evaluate the detection performance across different watermark generation methods under the three corruption types, as shown in Figure 4. Based on the experimental results, we draw the following conclusions. First, in terms of the median p-value and empirical power graphs, we find that watermarks generated using the EXP-edit method maintain high robustness even when the perturbation rate reaches 0.7. Furthermore, across different experiments, we find that both ITS-edit and EXP-edit consistently

perform better than their non-edit counterparts (ITS and EXP). This is because EXP(ITS)-edit incorporates a Levenshtein-based alignment strategy during detection, which allows it to tolerate token substitutions, insertions, and deletions—making it significantly more robust to text corruptions compared to the original EXP(ITS) method[8]. Second, the null rejection rate remains consistently around 0.05, indicating that the false positive rate is well controlled. Additionally, the number of permutations required for detection remains significantly lower than that of traditional permutation tests, demonstrating the efficiency advantages of using SMC-based testing.



(a) Median p-value

(b) Empirical Power

(c) Null Rejection Rate

(d) Number of Permutation

Figure 4: Performance Evaluation under Different Watermark Generation Method

7 Conclusion

In this work, we present a robust and scalable Sequential Monte Carlo (SMC) testing framework for watermark detection in text. Our approach overcomes the practical limitations of traditional permutation tests by enabling early stopping once sufficient statistical evidence has been accumulated. Experimental results show that the SMC method retains high statistical power while substantially reducing computational costs—often requiring fewer than 50 permutations compared to the 1,000 typically needed by standard, non-sequential methods.

Through extensive evaluations, we demonstrate that the SMC test consistently outperforms traditional permutation testing across a wide range of conditions. In settings with limited samples or corrupted text, the SMC method exhibits enhanced robustness and reliability. It maintains strong statistical validity, effectively controls false positive rates, and achieves high empirical power in detecting watermarks. Notably, the method remains effective even under realistic token-level corruptions, including substitutions, insertions, and deletions, emphasizing its practicality for deployment in noisy or adversarial environments.

The core innovation of this project lies in replacing brute-force permutation testing with a sequential framework that dynamically adapts to the accumulating evidence. This enables both computational efficiency and rigorous statistical guarantees, establishing our approach as a promising solution for reliable watermark detection in modern natural language processing systems.

Individual Contributions

Devashish Juyal: Did literature survey of all kinds of watermarking methods then developed the sequential Monte Carlo testing framework for LLM watermarking (Sections 2–4, Appendices A–B), implemented detection algorithms, proposed experiments for power and null rejection evaluation, and analyzed the statistical validity of the test.

Emilio Cantu-Cervini: Assisted in the selection of the sequential test and its implementation. Performed power curve experiments (Figure 1) and aided their interpretations.

Zicheng Jin: Implement code to perform the SMC test for detection. Designed and conducted comprehensive experiments on the corruption test, visualize the results in Figures 3 and 4, and performed in-depth analysis.

Sri Likhita Adru: Implemented the designed the SMC algorithm into executable code. Conducted comparative analyses between the fixed-sample and sequential approaches. Aided with the robustness testing framework by evaluating the detection methods under deletion and substitution attacks, and analyzed their performance.

References

[1] Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*, 2019.

[2] Can Chen and Jun-Kun Wang. Online detecting llm-generated texts via sequential hypothesis testing by betting. *arXiv preprint arXiv:2410.22318*, 2024.

[3] Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1125–1139. PMLR, 2024.

[4] Lasse Fischer and Aaditya Ramdas. Sequential monte-carlo testing by betting. *arXiv preprint arXiv:2401.07365*, 2024.

[5] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[6] Steffen Herbold, Annette Hautli-Janisz, Ute Heuer, Zlata Kikteva, and Alexander Trautsch. A large-scale comparison of human-written versus chatgpt-generated essays. *Scientific reports*, 13(1):18617, 2023.

[7] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR, 2023.

[8] Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

[9] Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip Yu. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36, 2024.

[10] David Siegmund. *Sequential Analysis: Tests and Confidence Intervals*. Springer, 1985.

A Mathematical Background

In this section, we introduce the necessary mathematical concepts and definitions required to understand our project. We assume minimal prior knowledge and present rigorous definitions before describing how these ideas are instantiated in existing work.

Hypothesis Testing: A statistical hypothesis test involves two competing hypotheses: the null hypothesis H_0 and the alternative hypothesis H_1. A hypothesis test aims to determine whether observed data provides sufficient evidence to reject H_0 in favor of H_1.

Type I Error and Significance Level α: A Type I error is the probability of incorrectly rejecting H_0 when it is true. The significance level α specifies the maximum acceptable probability of a Type I error.

P-value: The p-value is the probability, under H_0, of observing data at least as extreme as the actual observation. If H_0 holds, the p-value is uniformly distributed on $[0, 1]$. We reject H_0 if the p-value satisfies

$$p \leq \alpha,$$

thus ensuring that the Type I error is controlled at level α.

Permutation Tests: Permutation tests are nonparametric tests that rely on the exchangeability of data under H_0. In a permutation test, one pools the data, applies random permutations to the labels or inputs, and recomputes a test statistic for each permutation. The permutation p-value is the fraction of permuted statistics that are at least as extreme as the observed statistic.

Because the null hypothesis implies that all labelings are equally likely, the permutation p-value is exact and distribution-free. When exhaustive permutation is computationally infeasible, Monte Carlo sampling of random permutations is used to approximate the null distribution.

Sequential Hypothesis Testing: Traditional hypothesis tests assume a fixed sample size. In contrast, sequential hypothesis testing allows observations to arrive one at a time, and the test may stop early once sufficient evidence is collected. However, naive sequential monitoring without adjustment can invalidate p-values and inflate the Type I error. Specialized methods, such as anytime-valid tests, are necessary to preserve Type I error control under optional stopping.

Testing by Betting Framework: The testing-by-betting framework reinterprets hypothesis testing as a sequential betting game against the null hypothesis. At each step, the tester places a "bet" on the next observation, updating a wealth process W_t based on the outcomes.

Under H_0, the wealth process W_t can be constructed as a nonnegative martingale satisfying

$$\mathbb{E}[W_t] = 1.$$

By Ville's inequality, the probability that W_t exceeds $1/\alpha$ at any time is at most α. Thus, rejecting H_0 when

$$W_t \geq \frac{1}{\alpha}$$

yields an anytime-valid test. Equivalently, the anytime-valid p-value process is given by

$$p_t = \frac{1}{\sup_{s \leq t} W_s}.$$

Martingales and E-Values: A stochastic process (M_t) is called a martingale if

$$\mathbb{E}[M_{t+1} \mid M_1, \ldots, M_t] = M_t.$$

A nonnegative martingale starting at $M_0 = 1$ satisfies $\mathbb{E}[M_t] = 1$ for all t under H_0.

An *e-variable* E is a nonnegative random variable satisfying

$$\mathbb{E}_{H_0}[E] \leq 1.$$

A sequence of e-variables generates an *e-process*. E-values and martingales are closely related: the wealth process W_t in a betting test forms an e-process, and

$$p_t = \frac{1}{W_t}$$

defines an anytime-valid p-value.

Pseudo-Random Number Generation in Sampling: Pseudo-random number generators (PRNGs) are deterministic algorithms that produce sequences of numbers resembling true randomness. A PRNG takes a secret seed and generates a reproducible pseudo-random sequence. In watermarking applications, a shared key ξ is used as a seed to drive deterministic sampling. Even though the outputs are reproducible given ξ, they appear random to external observers without access to the key.

Sequence Alignment and Alignment Cost: Sequence alignment is a method of arranging two sequences to highlight regions of similarity. Gaps can be inserted into either sequence, and penalties are assigned for mismatches and gaps.

The alignment cost is the total penalty incurred by aligning two sequences optimally. Algorithms such as Needleman–Wunsch, Smith–Waterman, and Levenshtein distance compute the optimal alignment using dynamic programming.

In our context, sequence alignment between a generated text and a key sequence helps detect the presence of a watermark, even when the text is corrupted by insertions, deletions, or substitutions.

Sequential Monte Carlo Testing: Fischer and Ramdas [4] developed sequential permutation tests using betting strategies. Instead of fixing a number of resampled permutations, their method draws samples sequentially, updating a wealth process W_t at each step.

Each new permutation contributes to growing W_t according to a betting strategy. By Ville's inequality, W_t remains a nonnegative martingale under the null hypothesis, ensuring that the sequential test remains valid at any stopping time.

The p-value at time t can be computed as

$$p_t = \frac{1}{\sup_{s \leq t} W_s},$$

maintaining anytime validity. This sequential Monte Carlo testing framework reduces computational burden by allowing early stopping once sufficient evidence against the null has been accumulated.

Thus, our project integrates distortion-free watermark generation from Kuditipudi et al. [8] with sequential, computationally efficient detection inspired by Fischer and Ramdas [4].

B Betting Functions in Sequential Permutation Testing

In a sequential Monte Carlo (permutation) test, random permutations are drawn one-by-one with the possibility of early stopping. Fischer and Ramdas propose treating this as a "betting" game: a gambler starts with unit wealth and bets each round on whether the newly generated test statistic Y_t is greater than the original Y_0. This approach yields an *anytime-valid* test: it remains valid regardless of when sampling stops.

Earlier permutation tests, such as the Besag–Clifford method, guarantee validity only at a prespecified stopping time, whereas the betting approach constructs a *test martingale* (a nonnegative

martingale starting at one) whose running maximum can be turned into a valid p-value at any stopping time. Under the null hypothesis (exchangeability), no betting strategy can systematically grow wealth on average. Under a working alternative hypothesis (where Y_0 is stochastically larger than the permutations), carefully chosen betting functions can accumulate evidence against the null.

B.1 Construction of the Betting Function

At each round t, the gambler chooses a *betting function*

$$B_t : \{0, 1\} \to \mathbb{R}_{\geq 0},$$

where the outcome $I_t = \mathbf{1}\{Y_t \geq Y_0\}$ is revealed (1 if $Y_t \geq Y_0$, otherwise 0). The wealth after T rounds is updated multiplicatively as

$$W_T = \prod_{t=1}^{T} B_t(I_t),$$

with initial wealth $W_0 = 1$.

To ensure (W_t) forms a test martingale under the null, the bets must satisfy the *martingale constraint*:

$$\mathbb{E}_{H_0}[B_t(I_t) \mid I_1, \ldots, I_{t-1}] = 1.$$

Under exchangeability, the conditional probabilities are

$$\mathbb{P}(I_t = 1 \mid I_{1:t-1}) = \frac{L_{t-1} + 1}{t + 1}, \quad \mathbb{P}(I_t = 0) = \frac{t - L_{t-1}}{t + 1},$$

where $L_{t-1} = \sum_{s=1}^{t-1} I_s$ counts prior "losses" (i.e., times when $Y_s \geq Y_0$). Thus, the constraint simplifies to:

$$B_t(0) \cdot \frac{t - L_{t-1}}{t + 1} + B_t(1) \cdot \frac{L_{t-1} + 1}{t + 1} = 1.$$

Typically, under the working alternative hypothesis (where Y_0 tends to be larger), one prefers $B_t(0) > 1$ and $B_t(1) < 1$, thereby favoring wins ($I_t = 0$) to grow wealth.

B.2 Null and Alternative Hypotheses

The null hypothesis H_0 asserts that $Y_0, Y_1, Y_2, \ldots$ are exchangeable. Thus, the rank of Y_t among $\{Y_0, \ldots, Y_t\}$ is uniform. The working alternative H_1 posits that Y_0 is stochastically larger than its permutations, leading to more "wins" ($I_t = 0$).

B.3 Wealth Process and E-Process

The wealth process (W_t) is a nonnegative martingale under H_0. Consequently, it is an *e-process*: for any stopping time τ,

$$\mathbb{E}_{H_0}[W_\tau] \leq 1.$$

Intuitively, under H_0, no betting strategy can systematically grow wealth, whereas under H_1, a well-chosen strategy may result in increasing wealth, providing evidence against H_0.

B.4 Ville's Inequality and Anytime-Valid p-Values

Ville's inequality states that, for any $\alpha \in (0, 1)$,

$$\mathbb{P}_{H_0}\left(\sup_{t\geq 1} W_t \geq \frac{1}{\alpha}\right) \leq \alpha.$$

Thus, one may reject H_0 at level α the first time $W_t \geq 1/\alpha$. Equivalently, an *anytime-valid p-value* process can be defined as

$$Q_t = \frac{1}{\max_{1\leq s\leq t} W_s},$$

so that for any stopping time τ, Q_τ is a valid p-value.

B.5 Standard Betting Strategies

Several betting strategies have been proposed:

- **Passive strategy**: Set $B_t(0) = B_t(1) = 1$ for all t, yielding no change in wealth.
- **Aggressive strategy**: Set $B_t(1) = 0$ and $B_t(0) = (t+1)/(t - L_{t-1})$, betting everything on a win. This recovers and generalizes the Besag–Clifford stopping rule.
- **Binomial strategy**: Introduce an aggressiveness parameter $\lambda \in (0, 1)$ to balance bets between wins and losses.
- **Binomial mixture strategy**: Average over multiple binomial strategies with different λ values drawn from a prior (e.g., uniform). The resulting wealth process remains a martingale and provides smoother, often more powerful behavior.

In all cases, the choice of B_t must satisfy the martingale constraint to guarantee validity. The wealth process thus constructed provides a flexible and powerful method for sequential hypothesis testing with strong statistical guarantees.

C Implementation Details

In this section, we describe the overall test setup, starting with the working of our proposed algorithms and their theoretical foundations.

Let V denote the vocabulary, and let $p : V^* \to \Delta(V)$ be an autoregressive language model that maps a prefix string of arbitrary length to a distribution over the next token, with $p(\cdot \mid x)$ representing the conditional distribution given prefix $x \in V^*$. Let Ξ denote the space of possible watermark key sequences.

The setup proceeds as follows:

1. The language model (LM) provider shares a random watermark key sequence $\xi \in \Xi^*$ with the detector.
2. The user submits a prompt $x \in V^*$ to the LM provider.
3. The LM provider generates text $Y \in V^*$ via a watermarked generation process: $Y = \text{generate}(x, \xi)$.
4. The user publishes a text $Y_e \in V^*$, which may either be:
   (a) An edited version of the generated text Y, or

(b) Text independent of Y (e.g., text written from scratch).

5. The detector tests whether Y_e is watermarked — i.e., whether it depends on ξ — by computing a p-value $p = \text{detect}(Y_e, \xi)$ under the null hypothesis that Y_e is independent of ξ (i.e., exchangeable).

For clarity, we divide the work into four core algorithms described below.

C.1 Watermarked Text Generation

The `generate` method constructs a watermarked text sequence by incorporating the watermark key into the decoding process. Given a watermark key sequence $\xi \in \Xi^n$, a language model p, and a decoding strategy Γ, it produces a text $y \in V^m$. At each step i, the decoder samples

$$y_i \leftarrow \Gamma\big(\xi_{(i \bmod n)},\, p(\cdot \mid y_{1:i-1})\big).$$

Algorithm 1 Watermarked Text Generation (`generate`)

Input: watermark key sequence $\xi \in \Xi^n$
Params: generation length m, model p, decoder Γ
Output: watermarked text $y \in V^m$

for $i = 1$ to m **do**
  $y_i \leftarrow \Gamma\big(\xi_{(i \bmod n)},\, p(\cdot \mid y_{1:i-1})\big)$
end for
return y

C.2 Watermarked Text Detection

The `detect` method computes $\Phi_0 = \Phi(y, \xi)$ on the candidate y, then for $t = 1, \ldots, T$ draws $\xi^{(t)} \sim \nu$ and $\Phi_t = \Phi(y, \xi^{(t)})$, and finally feeds $\{\Phi_0, \ldots, \Phi_T\}$ into the SMC test to get a p-value.

Algorithm 2 Watermarked Text Detection (`detect`)

Input: string $y \in V^*$, watermark key sequence $\xi \in \Xi^n$
Params: test statistic ϕ; watermark key sequence distribution $\nu \in \Delta(\Xi^n)$; resample size T
Output: p-value $p_b \in [0, 1]$

$\Phi_0 \leftarrow \Phi(y, \xi)$
for $t = 1$ to T **do**
  draw $\xi^{(t)} \sim \nu$
  $\Phi_t \leftarrow \Phi(y, \xi^{(t)})$
end for
run SMC test on $\{\Phi_0, \ldots, \Phi_T\}$ to obtain wealth $\{W_t\}_{t=1}^{\tau}$
$p \leftarrow 1/\max_{1 \le s \le \tau} W_s$
return p

C.3 Sequential Monte Carlo Test

We maintain wealth W_t under a binomial-mixture strategy with parameter $c < \alpha$:

$$W_0 = 1, \quad L_t = \sum_{i=1}^{t} \mathbf{1}\{\Phi_i \ge \Phi_0\}, \quad W_t = \frac{1 - \text{Bin}(L_t;\, t+1, c)}{c}.$$

Stop at τ when $W_t < \alpha$ or $W_t \ge 1/\alpha$.

Algorithm 3 Sequential Monte Carlo Test

Input: significance α; parameter $c < \alpha$; statistics $\{\Phi_0, \Phi_1, \dots\}$
Output: stopping time τ; wealth sequence $\{W_t\}$

$W_0 \leftarrow 1$, $L_0 \leftarrow 0$
for $t = 1, 2, \dots$ **do**
  if $\Phi_t \geq \Phi_0$ **then**
    $L_t \leftarrow L_{t-1} + 1$
  else
    $L_t \leftarrow L_{t-1}$
  end if
  $W_t \leftarrow \frac{1-\mathrm{Bin}(L_t;\, t+1,\, c)}{c}$
  if $W_t < \alpha$ **or** $W_t \geq 1/\alpha$ **then**
    return $\big(t, \{W_s\}_{s=1}^{t}\big)$
  end if
end for

C.4 Test Statistic

Define

$$\Phi(y, \xi) = \min_{1 \leq i \leq |y|-k+1} \min_{1 \leq j \leq n} d\big(y_{i:i+k-1},\, \xi_{j:j+k-1}\big),$$

where d is a block-alignment cost. Smaller Φ means stronger watermark alignment.