Sandcastles in the Storm: Revisiting the (Im)possibility of Strong Watermarking

Anonymous ACL submission

Abstract

Watermarking AI-generated text is critical for combating misuse. Yet recent theoretical work argues that any watermark can be erased via random walk attacks that perturb text while preserving quality. However, such attacks rely on two key assumptions: (1) rapid mixing (watermarks dissolve quickly under perturbations) and (2) reliable quality preservation (automated quality oracles perfectly guide edits). Through large-scale experiments and human-validated assessments, we find **mixing is slow**: 100% of perturbed texts retain traces of their origin after hundreds of edits, defying rapid mixing. **Oracles falter**, as state-of-the-art quality detectors misjudge edits (77% accuracy), compounding errors during attacks. Ultimately, **attacks underperform**: automated walks remove watermarks just 26% of the time – dropping to 10% under human quality review. These findings challenge the inevitability of watermark removal. Instead, practical barriers – slow mixing and imperfect quality control – reveal watermarking to be far more robust than theoretical models suggest. The gap between idealized attacks and real-world feasibility underscores the need for stronger watermarking methods and more realistic attack models.[1]

1 Introduction

The rapid proliferation of generative AI has created an urgent need for mechanisms to authenticate machine-generated content. Watermarking – embedding statistical signals into AI outputs to verify provenance – serves as a vital safeguard against misinformation, IP theft, and academic fraud. While traditional methods employ visual patterns (e.g., pixel-level changes in images), statistical watermarking for text encodes imperceptible signals at lexical or semantic levels through specially selected patterns of tokens (Liu et al., 2024b).

However, recent work by Zhang et al. (2024) ("Watermarks in the Sand," WITS) challenges the viability of watermarking, asserting that any such scheme can be defeated without degrading output quality through a simple random walk attack (see also, e.g., Kirchenbauer et al. (2024); Kuditipudi et al. (2024); Krishna et al. (2023)). This impossibility result threatens to undermine the accountability and security of generative AI, leaving no viable path to enforce ethical standards or trace misuse.

The text-based WITS attack employs two primary components: (1) a perturbation oracle $\mathbf{P}$ that iteratively modifies text, and (2) a quality oracle $\mathbf{Q}$ to ensure that the edits are reasonable. These induce a random walk on a (potentially enormous) graph $\mathbf{G}$, where nodes represent possible texts y and edges denote size-bounded perturbations (e.g., single-word swaps or paraphrases). Under certain assumptions, the random walk converges to a stationary distribution – a stable equilibrium over nodes that remains unchanged under further perturbations. Crucially, this stationary distribution is a function of $\mathbf{P}$ and therefore independent of any particular watermarking scheme. As the random walk approaches this equilibrium, the likelihood of encountering a $\mathbf{Q}$-approved unwatermarked text increases. Notably, the WITS attack prioritizes quality equivalence over semantic equivalence: it seeks unwatermarked texts that score similarly under $\mathbf{Q}$, even if their meaning diverges significantly from the original.

While elegant in theory, the WITS argument relies on two key assumptions (KA) that warrant further scrutiny. Specifically, WITS assumes that:

KA1. The transition probabilities assigned to quality-preserving perturbations are high enough to ensure rapid mixing. Formally, this means that the second-largest eigenvalue (in absolute value) of the transition matrix is sufficiently close to zero to ensure rapid mixing ((Zhang et al., 2024), Theorem 5).

[1] Code and datasets released upon acceptance.

KA2. The quality oracle Q can reliably preserve output quality throughout the attack. But if **Q** is unreliable – either by admitting low-quality outputs or by blocking valid edits – the attack either fails to escape the watermark or produces low-quality outputs that are no longer competitive with the original.

Taken together, **KA1** is concerned with attack efficiency and **KA2** further requires that the results remain meaningfully close to the initial text quality. To investigate whether these assumptions hold in practice, we designed analyses carefully tailored to study each assumption. For **KA1**, acquiring the eigenvalues of the transition matrix is infeasible due to its intractable size. Instead, we approximate mixing behavior by testing whether random walks retain memory of their starting states. If the random walk efficiently mixes, perturbed texts should lose memory of their starting points, making them indistinguishable from those originating elsewhere in the graph. Conversely, if stationary mixing is slow, initial states should remain identifiable even after many perturbations.

For **KA2**, we crafted a dataset of perturbations annotated with human quality judgments and benchmarked a variety of automated oracles to determine their reliability. We then used the best oracle to guide the random-walk attacks and cross-checked the quality of the final perturbed texts to fairly estimate the robustness of several representative watermarking schemes – KGW (Kirchenbauer et al., 2023), SIR (Liu et al., 2024a), and Adaptive (Liu and Bu, 2024). Our approach therefore addresses three primary research questions:

RQ1. *Can stationary distributions for watermarking be reached under practical constraints?* Even after hundreds of perturbations, starting states remain 100% distinguishable, strongly suggesting that stationary distributions are not within efficient reach.

RQ2. *Are LLM-based quality oracles sophisticated enough to guide a random-walk attack?* The top-performing oracle attained an F1-score of 77.4%, leaving significant room for errors to accumulate during the attack. This suggests that current generative oracles do not conform to the widely held belief that "verification is easier than generation."

RQ3. *How effective are random-walk attacks in breaking watermarks when controlling for quality?* Our improved random-walk attacks – whether operating on a word, span, sentence, or document level – succeeded in erasing the watermarks only 26.1% of the time on average. After humans reviewed the perturbed texts to determine if quality was truly preserved, success dropped to an average 10.5%.

Overall, our findings demonstrate a disconnect between theoretical assumptions and practical realities. These findings highlight the trade-offs adversaries face: preserving quality necessitates minimal edits, but escaping detection requires riskier perturbations that compromise output quality. By bridging theoretical critique with empirical validation, this work challenges the inevitability of strong watermarking's failure and offers a path forward for developing robust watermarking techniques grounded in real-world constraints.

2 Background

In this section, we outline the main objects and assumptions that underpin our analysis, following (Zhang et al., 2024). For formal definitions and more details, refer to Appendix A.

Let $\mathbf{M}$ be a generative model mapping prompts $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$ according to a probability distribution. Let $\mathbf{Q} : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ be a function that returns a quality score for y as a response to prompt x. We assume that the adversary has oracle access to $\mathbf{Q}$. Notice that the watermarked model can be used as the quality oracle since we are not editing y using $\mathbf{Q}$, whether or not this is sufficient to approximate $\mathbf{Q}$ is the content of **KA2**.

Let $\mathbf{P} : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$ be a randomized *perturbation oracle* that generates an alternative response y' from an original response y for the same prompt x. For the attack to succeed, $\mathbf{P}$ must preserve the quality of y with constant nonzero probability $\epsilon_{\text{pert}} \in (0, 1]$ (Definition A.1).

Starting from a watermarked response y_0, we iteratively apply $\mathbf{P}$ to generate mutations by setting $y_i = \mathbf{P}(x, y_{i-1})$. To maintain high quality, each mutation must satisfy $\mathbf{Q}(x, y_i) \geq q$ (for some $q \in [0, 1]$); otherwise, it is rejected.

We now formalize the graph $\mathbf{G}_x^{\geq q}$ underlying the random walk induced by this process as the graph whose nodes are the output space of $\mathbf{M}$ when given x as input such that $\mathbf{Q}(x, y) \geq q$, and whose edges are all pairs (y, y') such that $\Pr\big[y' = \mathbf{P}(x, y)\big] > 0$, with the weight of the edge given by $\Pr\big[y' = \mathbf{P}(x, y)\big]$ (Definition A.5).

To ensure the success of the WITS attack, we need to impose mixing assumptions on the random

walk, *irreducibility* (Definition $A.3$) and *aperiodicity* (Definition $A.4$). Together, these assumptions ensure that the random walk converges to a unique stationary distribution $\vec{\pi}$ (Definition A.2). In particular, after a sufficient number of steps, the probability of being at any node becomes independent of the initial state. This is critical for the WITS attack analysis: irreducibility guarantees that the random walk is not trapped within a single connected component, and aperiodicity prevents cyclic behavior that could hinder convergence. Given the enormous size of $\mathbf{G}$, aperiodicity is expected to hold. We discuss the irreducibility assumption further in Section 5.

We now define the mixing time of an irreducible and aperiodic graph:

Definition 2.1 ((Zhang et al., 2024), Definition 9)**.** Let $\mathbf{G} = (\mathcal{V}, \mathcal{E})$ be an irreducible and aperiodic weighted directed graph with transition matrix $\vec{P}$. For any $\epsilon_{\text{dist}} \in (0, 1]$, the ϵ_{dist}-*mixing time* $t_{\min}(\epsilon_{\text{dist}})$ of $\vec{P}$ is the smallest t such that for every starting distribution $\mathbf{p}_0 \in \mathbb{R}^n$, we have

$$|\mathbf{p}_t - \vec{\pi}| = \left|(\vec{P}^\top)^t \cdot \mathbf{p}_0 - \vec{\pi}\right| \leq \epsilon_{\text{dist}},$$

where $\mathbf{p}_t$ denotes the distribution over the vertices after t steps.

After $t_{\min}(\epsilon_{\text{dist}})$ steps, with probability at least $1 - \epsilon_{\text{dist}}$, a sample drawn from the random walk behaves as if drawn from the stationary distribution – i.e. independent of the original watermarked text.

Moreover, the mixing time $t_{\min}(\epsilon_{\text{dist}})$ can be bounded in terms of the second largest eigenvalue g (in absolute value) of $\vec{P}$ and the minimum stationary probability $\pi_{\min} = \min\{\vec{\pi}(1), \ldots, \vec{\pi}(n)\}$

$$t_{\min}(\epsilon_{\text{dist}}) \leq O\left(\frac{1}{1-g} \cdot \log\left(\frac{1}{\pi_{\min} \cdot \epsilon_{\text{dist}}}\right)\right).$$

In practice, particularly for prompts with high entropy where the number of acceptable outputs (and hence the size of $\vec{P}$) is extremely large, estimating g and thus $t_{\min}(\epsilon_{\text{dist}})$ becomes challenging. This difficulty directly relates to **KA1** and underscores the adversary's challenge in determining when to halt the random walk. This is discussed further in Appendix H.

It is important to note that for an attack to be considered successful, the adversary A must be significantly weaker than the model $\mathbf{M}$. Otherwise, A could simply ignore the watermarked output y and generate a fresh answer to x, thereby trivially bypassing the watermark. Also notice that the step size of $\mathbf{P}$ directly impacts the mixing time of the random walk, which motivates the choice of our perturbation oracles.

At a high level, Theorem 2 in (Zhang et al., 2024) proves that if these mixing conditions are satisfied, the random walk attack breaks any watermarking scheme with running time proportional to $\frac{1}{1-g}$. Moreover, the attacker can control the trade-off between quality of the final unwatermarked text and the probability of removing the watermark.

3 Evaluation Setup

We now describe the main components of our evaluation: the watermarking schemes, the dataset, the automated quality metrics, and the perturbation oracles. We defer quality-oracle details to RQ2, where we benchmark and justify using `InternLM` as our primary $\mathbf{Q}$ in our attacks.

Watermarkers. We evaluate three widely used watermarking schemes $\mathbf{W}$: KGW (Kirchenbauer et al., 2023), SIR (Liu et al., 2024a), and Adaptive (Liu and Bu, 2024). Each embeds signals into generated text to enable authorship attribution. KGW utilizes a "red-green" list of tokens determined by the rolling hash of the previous k tokens (typically $k = 3$ to 5). The logit scores for "green" tokens are boosted slightly to promote their selection. SIR follows a similar structure but instead relies on the semantic embeddings of preceding tokens, making it a form of "semantic" watermarking. Adaptive restricts its modifications to tokens in high-entropy regions to preserve text quality while still embedding a watermark. Because SIR and Adaptive each incorporate semantic context, both qualify as semantic watermarking schemes designed to resist attacks that preserve meaning through paraphrase. We note that these watermarking schemes produce detection scores on different scales: some, such as Adaptive, use a 0–100 scale, whereas KGW and SIR compute a z-statistic. Additional details about the watermarkers can be found in Appendix B.1.

Dataset. As noted in Section 2, the number of valid responses to a prompt (i.e., its *entropy*) influences the structure of the perturbation graph $\mathbf{G}$. To systematically investigate the relationship between entropy and attack success, we constructed a dataset for **RQ1** and **RQ3** featuring entropy-controlled prompts in three domains relevant to au-

thorial accountability: education, journalism, and creative writing.

For each domain, we designed a series of prompts with increasing specificity. For instance, a broad request might be "Write a 500-word news article," while a more constrained one could read "Write a 500-word news article about a global climate summit"(see Appendix B.2 for more details). We used the `Llama-3.1-70B-Instruct` model (denoted as $\mathbf{M}$) to generate three watermarked texts per prompt for each of the three watermarking schemes, resulting in 270 watermarked texts. To provide a baseline, we also generated 90 unwatermarked texts from the same model. In addition, we included unwatermarked outputs from `GPT-4o` to represent scenarios with higher quality text that adversaries might favor. For each watermarking scheme, we computed the mean watermark detection score and its standard deviation on the unwatermarked texts to establish a reference range against which we measure whether a perturbed text remains distinguishable.

Automatic Quality Metrics. We automatically evaluated text quality with several metrics. First, we used `InternLM` (Cai et al., 2024) as a reward model to acquire a numerical quality score. We used GPT-2 (Radford et al., 2019) to calculate perplexity, with lower values indicating higher fluency and predictability. In addition, we counted the number of grammatical errors using a standard grammar checker (languagetool, 2023), and we computed the unique bigrams in each text to assess lexical diversity. Table 5 in Appendix B.3 summarizes these metrics for both unwatermarked and watermarked texts before attack.

Perturbation Oracles Our perturbation oracles, denoted as $\mathbf{P}$, apply adversarial edits at different levels of granularity. `WordMutator` and `EntropyWordMutator` replace individual tokens, with the latter concentrating on high-entropy tokens that are most likely to carry stronger watermark signals. `SpanMutator`, which is the primary text-based perturbation approach used by Zhang et al. (2024), masks $n = 6$ contiguous tokens and refills them using `T5-XL` (Raffel et al., 2019). `SentenceMutator` paraphrases a single sentence at each step with `Llama-3.1-8B`, while `DocumentMutator`, `Document1StepMutator`, and `Document2StepMutator` attempt a full-document paraphrase, either in one pass or in multiple passes. We allow a sufficient number of steps (e.g., 1000 for token-level edits and 100 for document-level edits) to maximize the opportunity for watermark removal. Further technical details on each $\mathbf{P}$ are provided in Appendix B.4.

4 Analysis

We now empirically examine the WITS assumptions by asking: (RQ1) whether the random-walk attacks reach a stationary distribution, (RQ2) whether LLM-based quality oracles reliably guide the attack, and (RQ3) how often watermark removal preserves text quality.

4.1 RQ1. Can Stationary Distributions Be Reached Under Practical Constraints?

WITS posits that repeated perturbations rapidly decouple a text from its starting state, eventually sampling from a stationary distribution. If mixing is slow, however, watermark removal becomes impractical in real-world applications. Although the second-largest eigenvalue (g) of the transition matrix provides a formal measure of mixing speed, computing g directly is infeasible due to the high dimensionality of $\mathbf{G}$. Instead, we measure whether the final texts can be traced back to their initial forms. If this tracing remains accurate even after many perturbations, it suggests that the random walk has not mixed sufficiently.

To approximate mixing, we propose a novel *lineage distinguisher* test. First, we choose two initial responses for each prompt. We then run a random-walk attack, perturbing one starting text until a $\mathbf{P}$-specific step budget is used up (e.g., 1000 steps for `WordMutator`). Periodically sampled texts along this walk are then classified by `Llama-3.1-70B-Instruct`, which attempts to identify their true origin. Since well-mixed texts should be indistinguishable from random samples in $\mathbf{G}$, classification accuracy should collapse to chance if a stationary distribution is reached.

4.1.1 Results

Table 1 summarizes results of a multi-stage classification approach designed to balance accuracy and cost. We first use `Llama-3.1-70B-Instruct` with a zero-shot prompt in a best-of-2 (see Appendix C for details). If `Llama-3` produces a tied result (considered a failure), we escalate to the stronger (but more expensive) `GPT-4o` (OpenAI, 2024b). Any remaining cases are then passed to `o3-mini-high` (OpenAI, 2025). At no point were both trials wrong in a best-of-2. Across all tests,

P Oracle	Steps	Tests	Llama-3.1-70B	GPT-4o	o3-mini-high
Word	1000	720	0	0	0
EntropyWord	1000	720	0	0	0
Span	250	720	12	1	0
Sentence	150	720	38	3	0
Document	100	421	2	0	0
Document1Step	100	576	0	0	0
Document2Step	100	678	1	0	0
Total / Failed Tests		**4555**	**53**	**4**	**0**
Cumulative Distinguished (%)			**98.84**	**99.91**	**100**

Table 1: Summary of failed distinguisher tests per **P**, along with the step budget and total tests. Classification is first performed by Llama-3.1-70B, followed by GPT-4o on its failures, then o3-mini-high on any remaining cases. The overall 100% success rate indicates that the attacked texts never lose memory of their starting points, contradicting **KA1** and suggesting that a stationary distribution is not reached in practice.

Llama-3.1-70B-Instruct alone achieves 98.84% accuracy. GPT-4o correctly resolves nearly all of the remaining 53 failures, and o3-mini-high succeeds on the last four, yielding a final 100% success rate. This consistently high distinguishability shows that random walks do not adequately mix within the allotted steps, thus contradicting **KA1** and indicating that the attacked texts remain too similar to their originals for watermark removal to rely on a converged stationary distribution.

4.2 RQ2. Are LLM-based quality oracles sophisticated enough to guide a random-walk attack?

A core assumption of WITS-style attacks is that verifying output quality is at least as easy as generating content ((Zhang et al., 2024), §4.1.2). This assumption aligns with the common belief that recognizing high-quality work – whether in music, cinema, or literature – is simpler than creating it. However, this premise has not been rigorously tested in the context of generative LLMs. If **Q** is unreliable – either by approving degraded outputs or blocking valid transformations – the attack stalls or yields low-quality text. To examine this assumption systematically, we built and benchmarked a variety of LLM-based oracles, measuring their ability to preserve quality while guiding watermark removal.

The Sandcastles Benchmark. We created the Sandcastles dataset to evaluate oracle reliability by sampling 100 diverse prompts from arena-human-preference-55k (Chiang et al., 2024), generating watermarked responses, and applying up to 20 iterative perturbations. At the 1st, 10th, and 20th steps, we collected human annotations comparing the perturbed text to its original. To ensure unbiased evaluation, annotators were presented with two texts, A and B, without knowing which had been perturbed. They provided ternary preference judgments, selecting either A, B, or tie.

For oracle training and evaluation, we binarized judgments: preferences for the perturbed text or a tie were labeled as "Quality Preserved," while preferences for the original were labeled as "Degraded." This simplification provides a clearer evaluation signal while preserving human preference patterns. The final dataset includes 795 annotated perturbations, with additional statistics in Appendix E.2.

Constructing and Evaluating Oracles. As a baseline, we followed the WITS suggestion to reuse the watermarking model **M** (Llama-3.1-70B-Instruct) as a quality oracle. After initial trials revealed positional biases and inconsistencies with human judgments, we explored several improvements. We ran oracle queries multiple times with flipped text orders, explicitly explained that the task involved assessing mutation quality (MutationOracle), and supplemented prompts with a changelog of all edits (DiffOracle). We then fine-tuned the strongest of these oracles on the Sandcastles training set (MutationOracle+FT, DiffOracle+FT). In parallel, we evaluated six reward models from the RewardBench leaderboard,[2] each producing continuous scores that we thresholded (e.g., a 0.46 deviation from the original in InternLMOracle) to classify outputs as high-quality or degraded. Finally, even though cost concerns make large proprietary models impractical for full attacks, we tested GPT-4-Turbo, GPT-4o, and a fine-tuned version GPT-4o+FT to gauge whether more powerful models offer significant improvements. Ad-

[2] https://huggingface.co/spaces/allenai/reward-bench

Oracle	Model	QP Prec.	Overall F1
MutationOracle	Llama-3.1-70B-Instruct (Dubey et al., 2024)	**84.62**	66.93
Prometheus2Absolute	GPT-4-Turbo (OpenAI, 2024a)	76.15	67.55
InternLMOracle	internlm2-20b-reward (Cai et al., 2024)	65.69	69.84
DiffOracle	Llama-3.1-70B-Instruct (Dubey et al., 2024)	71.74	70.85
DiffOracle+FT	Llama-3.1-70B-Instruct + Fine-tuning	69.07	76.94
MutationOracle+FT	GPT-4o (OpenAI, 2024b) + Fine-tuning	74.51	**77.38**

Table 2: Performance of selected quality oracles on human-annotated data (full results in Appendix E.1). QP Precision measures accuracy in preserving high-quality outputs, while Overall F1 reflects general classification performance. Despite fine-tuning, no oracle fully aligns with human judgments, challenging **KA1** and limiting their reliability in guiding random-walk attacks.

ditional details on these oracle variants appear in Appendix E.1.

We report both Quality Preserved (QP) Precision and Overall F1 to assess oracle performance. High QP Precision reduces false-positive approvals of degraded texts, a critical safeguard against cumulative quality erosion during multiple perturbations. The Overall F1 captures an oracle's overall ability to classify text quality accurately.

4.2.1 Results

Table 2 summarizes each oracle's runtime, return type, and performance. Our results show that current LLM-based quality oracles remain inconsistent, limiting the feasibility of using them to guide watermark removal attacks. Even the best-performing oracle (`GPT-4o+FT`) attains an Overall F1 of only 77.4%, implying that nearly one in five perturbations is misclassified. Fine-tuning and the use of powerful models like `GPT-4o` and `GPT-4-Turbo` do reduce errors somewhat, but not to a level sufficient for reliably guiding multi-step attacks. This misclassification compounds over repeated perturbations, forcing adversaries to either accept noticeable quality loss or proceed with low attack efficiency.

Among locally hosted models, the most robust approaches used difference-aware or mutation-aware prompts – `DiffOracle` (QP Precision: 70.9%) and `MutationOracle` (QP F1: 66.9%). Even after fine-tuning, however, these oracles still frequently labeled degraded outputs as high-quality. Moreover, high-scoring reward models from RewardBench (e.g., `INFORMOracle` at 95.1, `SkyworkOracle` at 94.3) often performed worse than simpler approaches, suggesting that generic reward tuning does not align well with the nuances of watermark-focused attacks. Collectively, these errors highlight a key limitation: LLM-based verification is not as reliable as assumed. We discuss potential causes for this limitations in Section 5.

Extended Comparison. While `GPT-4o+FT` achieves the best results overall, its high cost makes it impractical for many-step attacks. We therefore ran a human evaluation comparing two locally hosted oracles – the best boolean-based (`DiffOracle+FT`) and the best floating-point (`InternLMOracle`) – in a 150-step attack using `SentenceMutator`, which induced the most mixing in **RQ1**. Human judges found that `InternLMOracle` preserved quality in 47.78% of samples, compared to 40.0% for `DiffOracle+FT` (Table 12). Bayesian analysis indicated an 85.08% probability that `InternLMOracle` is genuinely superior (Appendix E.4), leading us to select `InternLMOracle` for further experiments despite its remaining error rate.

4.3 RQ3. How effective are random walk attacks in breaking watermarks when controlling for quality?

Attack Methodology. We apply various perturbation oracles to texts watermarked by KGW, SIR, or Adaptive. At each step, a candidate edit is proposed and accepted only if our quality oracle (`InternLMOracle`) labels it as high-quality. We track watermark detection scores and terminate when a fixed number of mutation steps is reached (details in Appendix G). An attack is deemed successful if the final detection score is less than $\mu_{\text{uw}} + 2\sigma_{\text{uw}}$, where μ_{uw} and σ_{uw} are, respectively, the mean and standard deviation of unwatermarked texts' detection scores. Under the assumption that scores follow a normal distribution, being below this threshold places the text in a region where fewer than 2.3% of unwatermarked samples lie above it, making it highly unlikely to be flagged as watermarked. We define the *attack success rate* (ASR) as the proportion of final texts that satisfy this criterion. We record two key states along each attack trace: (1) s_{min}, corresponding to the lowest watermark score achieved (as if an attacker had real-

W	P Oracle	μ_{w_0}	μ_{w_t}	ASR_{min}	ASR_{fin}	Reviewed	QP	¬QP	$\text{Q-ASR}_{\text{fin}}$
Adaptive	Word	99.27	70.37	0.00	0.00	0	0	0	0.00
Adaptive	EntropyWord	99.27	82.45	0.00	0.00	0	0	0	0.00
Adaptive	Span	99.27	67.21	1.54	1.54	2	2	0	1.54
Adaptive	Sentence	99.27	59.93	35.34	19.21	20	8	12	7.68
Adaptive	Document	99.27	58.55	48.78	45.24	20	8	12	18.10
Adaptive	Document1Step	99.27	70.94	1.16	1.16	2	2	0	1.16
Adaptive	Document2Step	99.27	73.39	5.33	4.71	8	5	3	2.94
SIR	Word	5.32	1.74	78.22	57.89	20	1	19	2.89
SIR	EntropyWord	5.32	3.30	39.68	27.54	20	0	20	0.00
SIR	Span	5.32	1.57	60.71	37.40	20	5	15	9.35
SIR	Sentence	5.32	0.52	87.65	74.71	20	13	7	48.56
SIR	Document	5.32	0.93	61.54	46.09	20	6	14	13.83
SIR	Document1Step	5.32	2.54	14.04	14.04	12	11	1	12.87
SIR	Document2Step	5.32	3.07	68.09	49.06	20	12	8	29.44
KGW	Word	0.28	0.17	47.54	20.00	20	4	16	4.00
KGW	EntropyWord	0.28	0.22	3.45	0.56	1	0	1	0.00
KGW	Span	0.28	0.20	38.46	32.35	20	14	6	22.65
KGW	Sentence	0.28	0.14	89.47	56.52	20	7	13	19.78
KGW	Document	0.28	0.18	62.50	44.44	20	8	12	17.78
KGW	Document1Step	0.28	0.27	12.66	8.54	14	7	7	4.27
KGW	Document2Step	0.28	0.18	9.09	7.78	10	4	6	3.11
Averages (%)				36.44	26.13		40.48	59.52	10.47

Table 3: Attack success rates (ASR) across different perturbation strategies. Human review reveals an average of 59.52% of successfully attacked texts have degraded quality. $\boldsymbol{\mu_{w_0}}$ represents the initial watermark score at step 0, while $\boldsymbol{\mu_{w_t}}$ represents the final watermark score after t mutation steps."min" refers to the point where the watermark score is at its lowest during the attack while "fin" refers to score at the final step of the attack. "Reviewed" indicates the number of human-reviewed examples where the watermark was broken. **QP** and **¬QP** represent the number of cases where human reviewers judged the attacked text as quality-preserving or degraded, respectively. $\textbf{Q-ASR}_{\text{fin}}$ is the re-estimated attack success after controling for quality, calculated as $\textbf{ASR}_{\text{fin}} \times (\text{QP}/\text{Reviewed})$.

time detector feedback), and (2) s_{fin}, produced when the perturbation budget is exhausted without direct detector feedback. This distinction clarifies how close attacks can come to fully erasing the watermark under ideal versus practical conditions.

Quality Evaluation. Since automatic metrics can overlook subtle forms of degradation, we also ran a human evaluation using 10 volunteer annotators, primarily with STEM backgrounds, but no specialized linguistic expertise. Each annotator compared a watermarked text with its final attacked version, randomly labeled as `text_A` or `text_B`, and chose whether one was superior or if they were equivalent. Instructions (Appendix F) emphasized fluency, style, coherence, and prompt relevance. We gathered up to 20 successfully attacked texts per watermark-oracle pairing, then determined how often annotators judged the attacked text to be at least as good as the original. This proportion was used to adjust the baseline $\text{Q-ASR}_{\text{fin}}$ value, giving a more realistic estimate of attack success.

4.3.1 Results

Table 3 shows that watermarks remain highly resistant to random-walk attacks, especially under quality constraints. Across all perturbation strategies, final attack success rates ($\textbf{ASR}_{\textbf{fin}}$) are low at 26.1% on average, with Adaptive proving the most robust. After human review, adjusted success rates ($\textbf{Q-ASR}_{\textbf{fin}}$) drop by an average of 59.52%, with no fairly evaluated attack succeeding more than half the time.

The effectiveness of attacks varies by perturbation type. Document-level edits perform best against Adaptive and KGW, while sentence-level perturbations show the highest success against SIR and KGW. Yet, even sentence-level mutations fail in most cases when quality is considered. Adaptive remains resistant to all strategies, especially entropy-controlled and word-level mutations, where no attack succeeds ($\textbf{Q-ASR}_{\textbf{min}} = 0.00$). Word-level perturbations generally fail across all watermarking schemes.

Overall, these findings highlight a major limitation for adversaries: while watermark removal is theoretically possible, preserving quality remains a significant obstacle. Even under ideal conditions – full perturbation budgets and access to detection feedback – successful attacks often fail to maintain human-perceived text quality.

5 Discussion

The Tricky Irreducibility Assumption. Another core assumption underlying the WITS framework that is difficult to evaluate empirically is that $\mathbf{P}$ induces an *irreducible* graph $\mathbf{G}_x^{\geq q}$. In other words, in theory, any high-quality text state is reachable from any starting point via a sequence of edits that all remain above the quality threshold q. However, this assumption is highly nontrivial, especially considering (a) the inherent limitations of $\mathbf{P}$, (b) the fact that edits are often local, and (c) the possibility that some transitions may necessarily involve brief dips below the threshold.

To see why irreducibility might fail in practice, consider two high-quality responses to a prompt asking for a story: assume that one is Star Wars and another is The Lord of the Rings (LOTR). For one to transform into the other *while remaining above the threshold*, there would need to be a sequence of high-quality intermediate texts that blend elements of both franchises. If our threshold q is stringent – say, requiring not just correct language but also stylistic consistency and thematic clarity – then many "blend" stages would likely be muddled or incoherent, causing the text to fall below q.

Hence, it is reasonable to suspect that the high-quality subgraph might contain distinct "islands" that cannot reach one another without temporarily leaving $\mathbf{G}_x^{\geq q}$. In fact, when humans write – one character at a time – they invariably pass through numerous low-quality states (partial words, half-formed sentences) before arriving at any one of the various ways of saying something of quality. Local edit operators, such as those that insert or delete single tokens or small chunks of text, face a similar risk: even a small disruption can degrade quality if the threshold is strict.

That said, irreducibility might still be recovered if we loosen our assumptions. For instance, we might allow momentary dips in quality during transitions so long as the process does not "get stuck" below q; or we could permit larger, more context-aware edits that can leap more cleanly between stylistic domains. In practice, these motivations led to the development of the `Document2StepMutator`, which aims to ensure that modifications are localized enough to avoid substantial quality degradation, yet also sufficiently broad to permit meaningful jumps. This design tries to strike a balance between remaining "near" high-quality states most of the time and retaining enough flexibility to move across different regions of the text space – ideally preventing the formation of disconnected "islands" of high-quality text.

Why do LLMs Struggle to Verify? While humans intuitively find verification easier than generation, this asymmetry may actually *reverse* for LLMs due to their probabilistic architecture and training paradigms. The core tension arises from LLMs' design as next-token predictors (Brown et al., 2020), which optimizes them for fluency over factual accuracy or logical rigor (Bender et al., 2021; Lin et al., 2021). Though techniques like chain-of-thought prompting (Wei et al., 2022) can simulate self-checking, the models remain fundamentally tuned to generate plausible continuations – not to verify them.

Compounding this, LLMs lack exposure to the iterative critique processes that shape human judgment. Trained on polished outputs (Dodge et al., 2021), they rarely encounter explicit revisions (e.g., drafts with margin notes like "this plot point contradicts Chapter 3") that teach cause-effect relationships between quality and text structure (Stiennon et al., 2020). Consequently, their "critiques" often reduce to surface-level heuristics (e.g., associating complex syntax with professionalism) rather than principled reasoning.

Whether verification is inherently harder for LLMs may hinge on whether "quality" is reducible to "likelihood." If not, their adeptness at generating fluent text may paradoxically hamper verification, as polished outputs mask subtle shortcomings (Bender et al., 2021), creating a hall-of-mirrors effect where plausibility is mistaken for truth.

6 Conclusion

Our findings reveal that watermark removal via random-walk attacks is far less certain than theoretical work suggests. Slow mixing and imperfect quality verification create significant real-world barriers. These insights invite deeper investigation: evolving watermark schemes could exploit the difficulty of consistent, high-quality edits, while attackers must grapple with the costs and risks of large-scale text manipulation. Our study also highlights the need for quality measures that align with human judgment, not surface features. Addressing these challenges – mixing speed, oracle reliability, and quality standards – will ensure watermarking remains viable against sophisticated attacks.

Limitations

While our findings highlight practical barriers to random-walk attacks, several limitations constrain their generalizability. First, we focus on three watermarking schemes (KGW, SIR, Adaptive) and specific perturbation oracles. Other schemes (Pan et al., 2024; Ren et al., 2024) and attack methods, especially those with advanced error-correction or alternative pathways (e.g., Rastogi and Pruthi (2024)), may yield different results.

Second, while human verification is critical to assessing attack success – a factor often overlooked in prior work – our findings rely on a small, potentially non-representative group of annotators. Broader user studies, richer datasets, and more diverse oracle designs are needed to validate our conclusions across varied scenarios, though such efforts would require significant resources.

Third, our analyses rely on LLM-based oracles fine-tuned for quality judgment, which still misclassify 20% of edits. Future breakthroughs in text evaluation – such as low-cost *reasoning* models (DeepSeek-AI et al., 2025) or specialized reward functions – could improve verification accuracy to the levels required to sustain viable attacks.

Finally, while we tested hundreds of perturbations, resource constraints limited exploration of arbitrarily large edit sequences. In theory, infinite steps might approach WITS's stationary distribution, but our results reveal substantial practical barriers. Computational costs further hinder scalability: `DocumentMutator` (based on `DIPPER` (Krishna et al., 2023)) took 213 seconds per attack step, rendering large-scale edits impractical.

References

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.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. 2024. Internlm2 technical report. *Preprint*, arXiv:2403.17297.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan

Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jesse Dodge, Maarten Sap, Ana Marasovic, William Agnew, Gabriel Ilharco, Dirk Groeneveld, and Matt Gardner. 2021. Documenting the english colossal clean crawled corpus. *CoRR*, abs/2104.08758.

Nicolai Dorka. 2024. Quantile regression for distributional reward models in rlhf. *arXiv preprint arXiv:2409.10164*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

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

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2024. 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. 2024. Robust distortion-free watermarks for language models. *Preprint*, arXiv:2307.15593.

languagetool. 2023. languagetool.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *CoRR*, abs/2109.07958.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2024a. A semantic invariant robust watermark for large language models. *Preprint*, arXiv:2310.06356.

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

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024c. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.

Yepeng Liu and Yuheng Bu. 2024. Adaptive text watermark for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

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

Scott Lundberg et al. 2022. Guidance. `https://github.com/guidance-ai/guidance`.

Yang Minghao. 2024. infly/INF-ORM-Llama3.1-70B · Hugging Face — huggingface.co. `https://huggingface.co/infly/INF-ORM-Llama3.1-70B`. [Accessed 11-02-2025].

OpenAI. 2024a. Gpt-4 turbo. Accessed: 2025-02-10.

OpenAI. 2024b. Gpt-4o system card. Accessed: 2025-02-10.

OpenAI. 2025. Openai o3-mini. Accessed: 2025-02-10.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. 2024. MarkLLM: An open-source toolkit for LLM watermarking. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 61–71, Miami, Florida, USA. Association for Computational Linguistics.

Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. 2024. Offsetbias: Leveraging debiased data for tuning evaluators. *Preprint*, arXiv:2407.06551.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Saksham Rastogi and Danish Pruthi. 2024. Revisiting the robustness of watermarking to paraphrasing attacks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18100–18110, Miami, Florida, USA. Association for Computational Linguistics.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2024. A robust semantics-based watermark for large language model against paraphrasing. *Preprint*, arXiv:2311.08721.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize from human feedback. *CoRR*, abs/2009.01325.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *EMNLP*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. 2024. Watermarks in the sand: Impossibility of strong watermarking for generative models. In *Forty-first International Conference on Machine Learning*.

A Appendix: Formal Definitions

In this section, we provide formal definitions of objects mentioned in Section 2 and elaborate on some definitions. As with the background section, most of these are directly from (Zhang et al., 2024). Let us begin by providing formal definitions of objects mentioned in Section 2.

Definition A.1 (ϵ_{pert}-Preserving Perturbation Oracle, (Zhang et al., 2024), Definition 6)**.** Let $\mathbf{P} : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$ be a randomized oracle that, given (x, y), outputs a new response y'. The oracle $\mathbf{P}$ is said to be ϵ_{pert}-preserving if for every $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,

$$\Pr\Big[\mathbf{Q}\big(x, \mathbf{P}(x, y)\big) \geq \mathbf{Q}(x, y)\Big] \geq \epsilon_{\text{pert}}.$$

Definition A.2 ((Zhang et al., 2024), Definition 8)**.** Let $G = (V, E)$ be a weighted directed graph, and $\vec{P}$ be the transition matrix of G. We say that $\vec{\pi} \in \mathbb{R}^n$ is a stationary distribution for $\vec{P}$ if: $\vec{P}^\top \cdot \vec{\pi} = \vec{\pi}$.

Definition A.3. A weighted directed graph $G = (\mathcal{V}, \mathcal{E})$ is **irreducible** if for any pair of vertices $u, v \in \mathcal{V}$, there exists a directed path from u to v with non-zero weight. In other words, there exists some $t \geq 1$ such that $\vec{P}^t(i, j) > 0$.

Definition A.4. A weighted directed graph $G = (\mathcal{V}, \mathcal{E})$ is **aperiodic** if the greatest common divisor of the lengths of all directed cycles in G is 1.

Let us now formally define the (hierarchically ordered) graph representations of $\mathbf{P}$ based on a prompt $x \in \mathcal{X}$ and the quality threshold $q \in [0, 1]$.

Definition A.5 ((Zhang et al., 2024), Definition 7)**.** Fix an arbitrary prompt $x \in \mathcal{X}$ and consider the graph $\mathbf{G}_x = (\mathcal{V}_x, \mathcal{E}_x)$ whose vertex set is the output space of $\mathbf{M}$ (i.e., $\mathcal{V}_x = \mathcal{Y}$) and whose edge set $\mathcal{E}_x$ consists of all pairs (y, y') such that

$$\Pr\big[y' = \mathbf{P}(x, y)\big] > 0.$$

We assign weights $w : \mathcal{E}_x \to [0, 1]$ to the edges by defining

$$w(y, y') = \Pr\big[y' = \mathbf{P}(x, y)\big].$$

Note that while the vertices of the graph are determined by the prompt $x \in \mathcal{X}$ and the watermarking model $\mathbf{M}$, the edges and their weights are determined solely by $\mathbf{P}$. Let us now incorporate quality into the graph representation. Let $\mathbf{G}_x^{\geq q}$ be the subgraph of $\mathbf{G}_x$ given by

$$\mathcal{V}_x^{\geq q} = \{y \in \mathcal{Y} \mid \mathbf{Q}(x, y) \geq q\},$$

$$\mathcal{E}_x^{\geq q} = \{(y, y') \in \mathcal{Y} \times \mathcal{Y} \mid \mathbf{Q}(x, y) \geq q, \mathbf{Q}(x, y') \geq q, \Pr\big[y' = \mathbf{P}(x, y)\big] > 0\},$$

Notice that we can carry the same weight assignment to this subgraph. Iteratively applying $\mathbf{P}$ on this graph and rejecting low-quality mutations produces a random walk where

$$\vec{P}_{(y,y')} = \Pr\big[y' = \mathbf{P}(x, y)\big].$$

Before presenting the WITS impossibility result, we formally define watermarking schemes and related notions.

Definition A.6 ((Zhang et al., 2024), Definition 3)**.** Let $\mathcal{M} = \{\mathbf{M}_i : X \to Y\}$ be a class of generative models with key space K. A secret-key watermarking scheme for $\mathcal{M}$ consists of two efficient algorithms:

- **Watermark**$(\mathbf{M})$: A randomized algorithm that, given a model $\mathbf{M} \in \mathcal{M}$, outputs a secret key $k \in K$ and a corresponding watermarked model $\mathbf{M}_k : X \to Y$.
- **Detect**$_k(x, y)$: A deterministic algorithm that, given a secret key $k \in K$, a prompt $x \in X$, and an output $y \in Y$, returns a bit $b \in \{0, 1\}$ indicating whether the watermark is present ($b = 1$) or absent ($b = 0$).

We now define the false-positive rate ϵ_{pos} of a watermarking scheme. Notice that if we sample $y \xleftarrow{\$} \vec{\pi}$, we expect that y is unwatermarked with probability ϵ_{pos}. Since for every good watermarking scheme we need ϵ_{pos} to be very small, this implies that outputs sampled from $\vec{\pi}$ will be unwatermarked with high probability, i.e. with probability approximately $1 - \epsilon_{\text{pos}}$.

Definition A.7 ((Zhang et al., 2024), Definition 4)**.** A watermarking scheme Π has a false positive ϵ_{pos}-rate if, for every model $M \in \{\mathbf{M}_i : X \to Y\}$, for every prompt $x \in X$, and for every output $y \in Y$,

$$\Pr[\text{Detect}_k(x, y) = 1] \leq \epsilon_{\text{pos}}.$$

Let us now define what it means for an adversary to break a watermarking scheme Π. Notice that one might consider weaker notions of breaking a watermarking scheme, but they will be implied by the WITS result.

Definition A.8 ((Zhang et al., 2024), Definition 5)**.** Let $\Pi = (\text{Watermark}, \text{Detect})$ be a watermarking scheme for a class of generative models $\mathcal{M} = \{\mathbf{M}_i : X \to Y\}$ with associated quality function $\mathbf{Q} : X \times Y \to [0, 1]$. We say that an adversary A ϵ-breaks Π if for every $M \in \mathcal{M}$, for every prompt $x \in X$, we have:

$$\Pr\left[\text{Detect}_k(x, y') = 0 \text{ and } \mathbf{Q}(x, y') \geq \mathbf{Q}(x, y) : y \xleftarrow{\$} \mathbf{M}_k(x),\ y' \xleftarrow{\$} A(x, y)\right] \ \geq\ \epsilon$$

where the probability is taken over $(k, \mathbf{M}_k)$ output by $\text{Watermark}(M)$ and the random coins of A.

We now introduce an additional technical definition that enables the attacker to trade off between output quality and attack success probability. Let $v \in [0, 100]$ denote the desired quality percentile. In other words, the attacker aims to produce an unwatermarked output whose quality falls within the top v-th percentile among all responses generated by $\mathbf{M}$ on a given prompt x. To formalize this, define the set of achievable quality scores as

$$\mathbf{Q}_{M,x} = \left\{q : \Pr[\mathbf{Q}(x, \mathbf{M}_k(x)) = q : (k, \mathbf{M}_k) \xleftarrow{\$} \text{Watermark}(M)] > 0\right\}$$

and let $q_{\mathbf{M},x}$ denote the v-th percentile of $\mathbf{Q}_{M,x}$. We then define the overall minimum quality threshold as

$$q_{\min} = \min_{\mathbf{M} \in \mathcal{M},\, x \in \mathcal{X}} \{q_{\mathbf{M},x}\}.$$

We now state the WITS impossibility result.

Theorem 1 ((Zhang et al., 2024), Theorem 6)**.** Let $\Pi = (\text{Watermark}, \text{Detect})$ be a watermarking scheme for a class of generative models $\mathcal{M} = \{\mathbf{M}_i : \mathcal{X} \to \mathcal{Y}\}$ with an associated quality function $\mathbf{Q} : \mathcal{X} \times \mathcal{Y} \to [0, 1]$. Let $\mathbf{P} : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$ be a perturbation oracle (defined over the same prompt space $\mathcal{X}$ and output space $\mathcal{Y}$ as the class $\mathcal{M}$) with the same associated quality function $\mathbf{Q} : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ as Π. For every non-watermarked model $\mathbf{M} \in \mathcal{M}$, for every prompt $x \in \mathcal{X}$, for every quality $q \in q_{\min}$, let $\vec{\pi}_{x,q}$ be the unique stationary distribution of the transition matrix $\vec{P}_{x,q}$ of $G_x^{\geq q}$. Let $n_{x,q} = |\mathcal{V}_x^{\geq q}|$, $\pi_{\min}^{(x,q)} = \min\{\vec{\pi}_{x,q}(1), \ldots, \vec{\pi}_{x,q}(n_{x,q})\}$ and g be the second largest eigenvalue of $\vec{P}_{x,q}$ in terms of absolute value. Let $t_{\text{err}} > 0$ be a tunable parameter. Let $t_{x,q}$ be the ϵ_{dist}-mixing time of $\vec{P}_{x,q}$, defined as follows:

$$t_{x,q} = \omega\left(\frac{1}{1-g} \cdot \log\left(\frac{1}{\pi_{\min}^{(x,q)} \cdot \epsilon_{\text{dist}}}\right)\right)$$

Assume the following holds:

1. The watermarking scheme Π has a false positive ϵ_{pos}-rate;
2. The perturbation oracle $\mathbf{P}$ is ϵ_{pert}-preserving;
3. For every non-watermarked model $\mathbf{M} \in \mathcal{M}$, for every prompt $x \in \mathcal{X}$, for every quality $q \in [q_{\min}, 1]$, the q-quality x-prompt graph representation $\mathbf{G}_x^{\geq q}$ of $\mathbf{P}$ is irreducible and aperiodic.

Then, there exists an oracle-aided universal adversary $A^{\mathbf{P}(\cdot,\cdot),\mathbf{Q}(\cdot,\cdot)}$ that ϵ-breaks Π by submitting at most t queries to $\mathbf{P}$ where

$$\epsilon = \left(1 - \frac{v}{100}\right)(1 - \epsilon_{\text{pos}})(1 - \epsilon_{\text{dist}})\left(1 - \sum_{k=0}^{t-t_{\text{err}}-1} \binom{t}{k} (\epsilon_{\text{pert}})^k (1 - \epsilon_{\text{pert}})^{t-k}\right),$$

and

$$t = \max_{x \in \mathcal{X}, q \in [q_{\min}, 1]} \{t_{x,q}\} + t_{\text{err}}.$$

By carefully tuning the parameter t_{err} and running the attack long enough so that ϵ_{dist} becomes negligibly small, the adversary can achieve a success probability close to

$$\left(1 - \frac{v}{100}\right)(1 - \epsilon_{\text{pos}}).$$

For example, targeting the median quality output (i.e., setting $v = 50$) restricts the adversary's success probability to roughly half of the maximum achievable rate.

B Appendix: Evaluation Setup

B.1 Watermark Details

For KGW and SIR, we use the implementations contained within the `MarkLLM` package[3] (Pan et al., 2024) with their default configurations. For Adaptive, we used the author's implementation[4] and due to initially poor results, experimented heavily with different configurations to find one that best balanced initial quality and detectability for `Llama-3.1-70B-Instruct`. The three tunable parameters we explored were `alpha`, which thresholds the amount of token entropy required to watermark it; `delta`, which controls the strength of boosting for watermarked tokens; and `delta_0`, which is the strength for watermarking the first $M = 50$ tokens, which are always watermarked. Our analysis lead us to use `alpha` = 2.0, `delta` = 1.5, `delta_0` = 1.0. Despite this extensive search, we still encountered intermittent issues with controlling for generation length. Since all texts were capped at a maximum of 1024 tokens due to fixed input sizes for various embedding models, some Adaptive responses were truncated mid-sentence, contributing to their unusually high number of grammatical errors as seen in Table 5.

By plotting the distributions for each quality metric in Figures 1 through 4, we noticed that Adaptive and SIR were vulnerable to producing highly distorted text with numerous quality issues. For example, a single Adaptive generation contained over 250 grammatical issues, largely due to inexplicable letter case alterations (e.g. "Over ThE nexT FEw dAYs, maggie partICIpaTed EnThusiasticALly I-n All ThE acTivities OffeRed aT WIllOW crEeEk..."). We did not regenerate bad responses because the distortions were a natural consequence of the watermarking algorithm itself, and regenerating them would obscure an important challenge to their real-world use. If the algorithm produces highly distorted text in some cases, then an attack is actually more likely to *repair* quality, rather than merely preserving it. At least some cases in our study fit this profile and the attack should be fairly credited even if it generally does not work for texts that start at a higher standard of quality.

[3] https://github.com/THU-BPM/MarkLLM
[4] https://github.com/yepengliu/adaptive-text-watermark

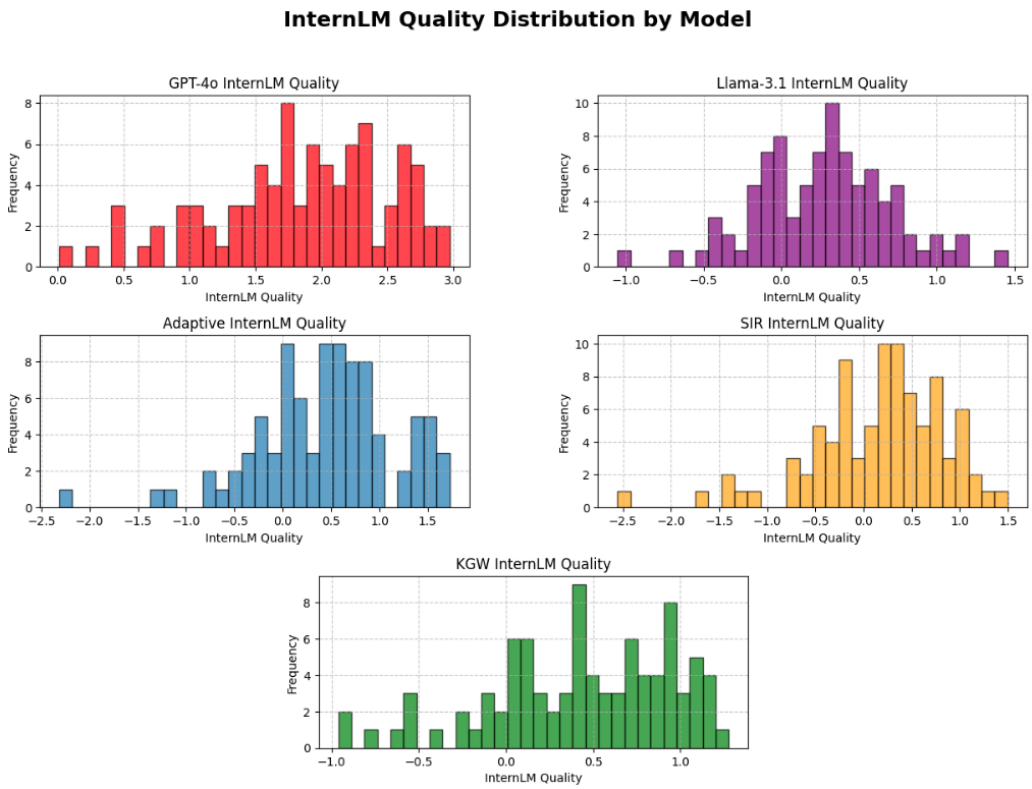

Figure 1: InternLM Quality Distribution by Watermarking Scheme $\mathbf{W}$

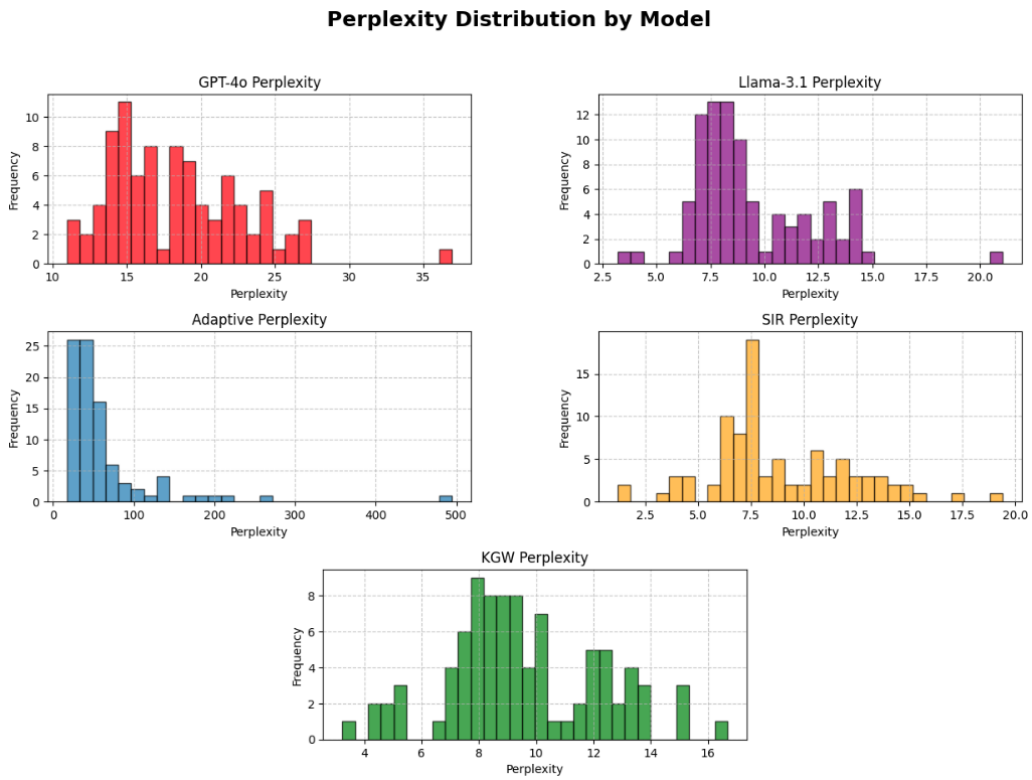

Figure 2: Perplexity Distribution by Watermarking Scheme $\mathbf{W}$

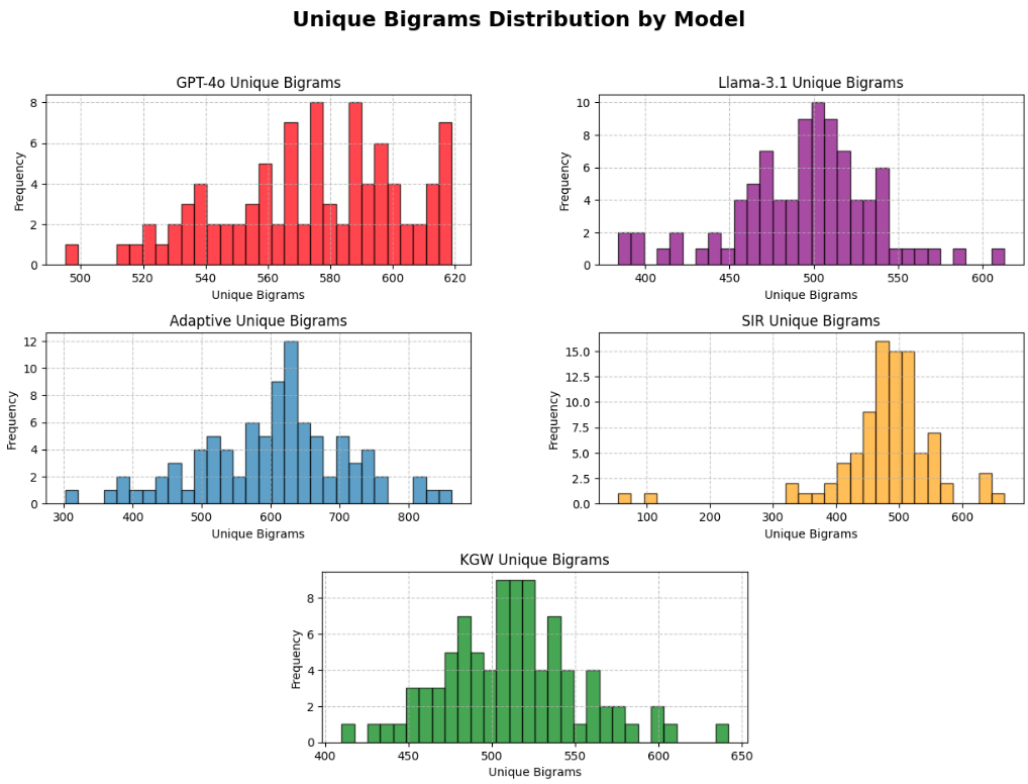

Figure 3: Unique Bigrams Distribution by Watermarking Scheme $\mathbf{W}$

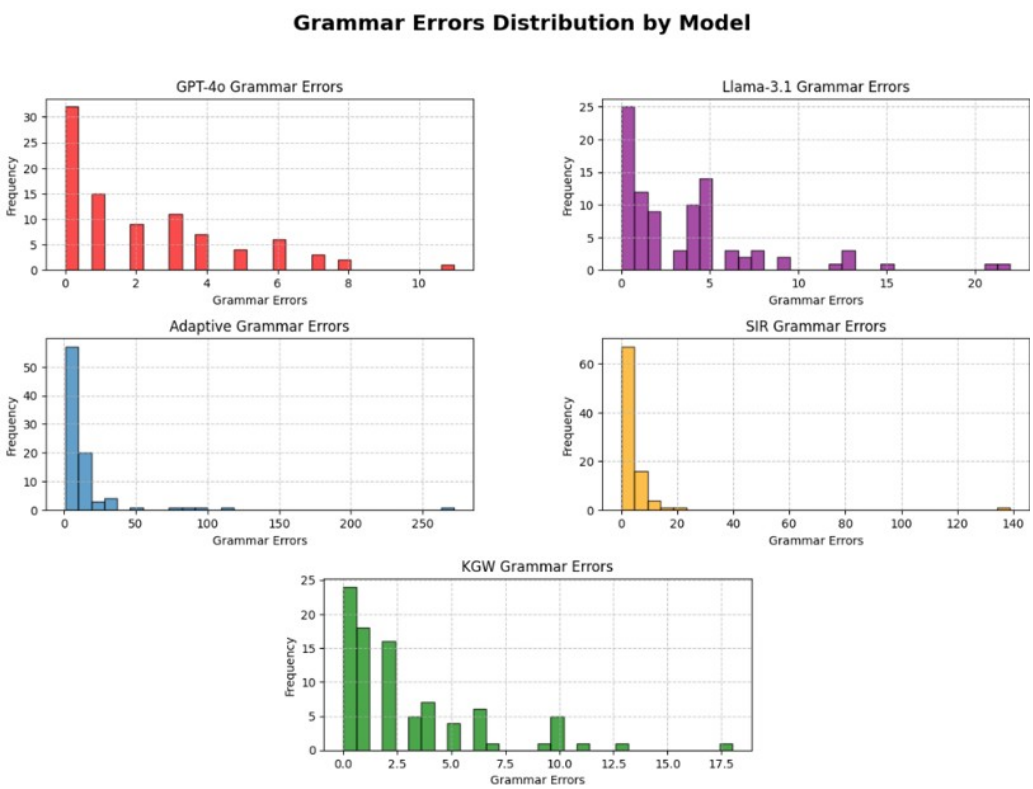

Figure 4: Grammar Errors Distribution by Watermarking Scheme $\mathbf{W}$

B.2 Entropy-Controlled Prompt Dataset

To systematically evaluate the impact of response entropy on watermark robustness, we curated a dataset featuring increasingly specific prompts across three domains: **creative writing**, **education**, and **journalism**. For each domain, we start with a broad, high-entropy prompt and progressively add constraints to reduce entropy. Below, we illustrate this progression with representative prompts at entropy level 1 (least constrained), 5, and 10 (most constrained).

Entropy Level	Prompt
Creative Writing	
1	Write a 500-word story.
5	Write a 500-word story about Evan, an American tourist, who falls for Emilie, a barista, during a spring festival in Paris.
10	Write a 500-word story about Evan, an American tourist, who falls for Emilie, a barista, during a spring festival in Paris. They bond over their love for Claude Monet's 'Impression, Sunrise' and the Hotel de Sully's architecture, leading to walks along the Seine. Their connection deepens amid shared laughter and explorations of Le Marais. As the festival lights dance on the river, Evan shares his feelings with Emilie under the starlit sky, promising to cherish the moments they've shared.
Education	
1	Write a 500-word essay about the importance of space exploration.
5	Write a 500-word essay about the importance of space exploration, its role in advancing human knowledge, and its potential to address global challenges like climate change and resource scarcity, with a focus on technologies developed for space missions.
10	Write a 500-word essay about the importance of space exploration, its role in advancing human knowledge, and its potential to address global challenges like climate change and resource scarcity, with a focus on technologies developed for space missions, their applications on Earth, the possibility of colonizing other planets like Mars, the ethical considerations of interplanetary exploration, and the cultural significance of humanity becoming an interstellar species.
Journalism	
1	Write a 500-word news article.
5	Write a 500-word news article about a global climate summit where world leaders are discussing strategies to combat climate change, with a focus on renewable energy investments and carbon reduction targets, highlighting a groundbreaking agreement between the US and China.
10	Write a 500-word news article about a global climate summit where world leaders are discussing strategies to combat climate change, with a focus on renewable energy investments and carbon reduction targets, highlighting a groundbreaking agreement between the US and China, featuring perspectives from small island nations affected by rising sea levels, addressing protests outside the summit calling for stronger climate justice measures, covering a controversial speech by a major oil industry representative, analyzing the summit's key outcomes and challenges, and placing it in the broader context of international efforts to achieve net-zero emissions by 2050.

Table 4: Representative entropy-controlled prompts across three domains: creative writing, education, and journalism. Entropy increases by adding specificity, progressively constraining the response space.

B.3 Dataset Statistics

	Unwatermarked		Watermarkered		
	GPT-4o	**Llama-3.1**	**Adaptive**	**SIR**	**KGW**
Mean Watermarked Score (μ_{w})	–	–	99.27	0.28	5.32
Mean Unwatermarked Score (μ_{uw})	–	–	49.43	0.08	-0.83
Unwatermarked Standard Deviation (σ_{uw})	–	–	3.37	0.07	1.05
Breakpoint (Score $\leq \mu_{\text{uw}} + 2\sigma_{\text{uw}}$)	–	–	56.16	0.21	1.27
Quality Score	1.85	0.27	0.45	0.16	0.43
Perplexity	18.39	9.38	63.32	8.87	9.56
Grammar Errors	2.20	3.69	16.21	4.24	2.86
Unique Bigrams Diversity	574.06	494.59	603.64	479.90	512.44
Mean Word Count	637.93	633.00	646.62	675.47	666.84
Generation Time (s)	15.24	274.27	671.75	335.12	292.24
Detection Time (s)	–	–	240.77	5.78	0.15

Table 5: Summary statistics for unwatermarked and watermarked text across different watermarking schemes, highlighting detection scores, automatic quality metrics, and runtime statistics.

B.4 Perturbation Oracle Details

The perturbation oracles $\mathbf{P}$ define the mechanism by which adversarial modifications are applied to watermarked text. These oracles generate perturbations of varying granularity, from token-level edits to full-document paraphrasing, enabling a systematic analysis of their impact on watermark robustness. Since prior work, including Zhang et al. (2024), has not accounted for how different perturbation strategies affect attack success, we explore a diverse set of perturbation oracles to quantify their relative effectiveness.

- **`WordMutator`**: Randomly replaces individual tokens by masking and filling them using `RoBERTa` (Liu et al., 2019).
- **`EntropyWordMutator`**: Similar to `WordMutator`, but uses `GPT-Neo-2.7B` (Black et al., 2021) to target high-entropy tokens for replacement as they are most likely to carry watermark signals.
- **`SpanMutator`**: Randomly masks six contiguous tokens at a time and fills them using `T5-XL` (Raffel et al., 2019). This is the only text-based perturbation oracle used in the WITS attack (Zhang et al., 2024).
- **`SentenceMutator`**: Randomly selects a sentence and paraphrases it creatively using `Llama-3.1-8B` (Dubey et al., 2024), introducing higher-level semantic shifts.
- **`DocumentMutator`**: Uses the `DIPPER` paraphrase model (Krishna et al., 2023) to paraphrase multiple sections of the document simultaneously.
- **`Document1StepMutator`**: Re-generates the entire document from scratch using `Llama-3.1-8B`, producing the most extreme form of perturbation while preserving meaning, quality, and formatting.
- **`Document2StepMutator`**: Performs a two-step transformation, first selecting a random sentence and paraphrasing it creatively with `Llama-3.1-8B`, then performs a global consistency editing to ensure that the remaining text is consistent with the edited sentence.

These perturbation oracles serve two key purposes in our study: (1) they enable us to analyze how the size of the perturbation affects movement within the perturbation graph $\mathbf{G}$; and (2) they allow us to determine whether specific perturbation oracles are more effective at breaking watermarks. By systematically evaluating these oracles, we aim to establish whether certain perturbation strategies inherently favor watermark removal and whether prior work has underestimated their impact on attack success.

To ensure sufficient opportunity for watermark removal, we allow a large number of perturbation steps, proportional to the average number of words edited per step. For example, WordMutator is given 1000 steps, while DocumentMutator is given 100. Additionally, we note that each perturbation oracle was carefully calibrated to balance subtle modifications with sufficient impact on watermark signals, ensuring reproducibility by fixing random elements such as token selection and sampling temperature. Table 6 reveals a clear trade-off: while fine-grained oracles tend to preserve fluency, coarse-grained methods introduce larger variations – a difference that is partly mitigated by the consistency editing in the `Document2StepMutator`.

$\mathbf{P}$	Steps	Edits	PPL ↓	Gram Err ↓	Approval ↑	Blocked ↓	QScore ↑	Time (s) ↓
`Word`	1000	1.8	40.4	10.2	0.80	0	-0.0688	**0.10**
`EntropyWord`	1000	1.1	16.8	9.2	**0.82**	0	-0.0949	0.28
`Span`	250	8.7	27.2	7.8	0.67	0	-0.0746	0.77
`Sentence`	150	31.3	21.0	4.6	0.74	0	0.2065	0.94
`Document`	100	216.0	11.2	9.1	0.36	0.12	-0.3151	213.12
`Document1Step`	100	138.2	**10.6**	**2.1**	0.42	0.12	0.0536	29.61
`Document2Step`	100	105.2	14.9	5.2	0.54	0.03	**0.2089**	34.78

Table 6: Performance metrics for each perturbation oracle. The columns report the number of attack steps, average edits per step, average text perplexity (PPL), average number of grammar errors, average $\mathbf{Q}$ approval rate, average rate at which $\mathbf{Q}$ blocks every perturbation for a given prompt, average `InternLM` quality score (QScore), and average runtime per perturbation step in seconds. Emboldened values denote the best performance per metric.

C Appendix: Distinguisher Prompt

Designing a distinguisher requires careful consideration, as failures can be difficult to attribute to either implementation flaws or inherent limitations due to sufficient mixing. To address this challenge, we dedicated significant effort to prompt engineering, developing 12 different distinguishers and systematically benchmarking their performance. Somewhat unexpectedly, one of the simplest approaches proved to be the most effective, and we describe it below.

An important implementation detail is ensuring that our distinguisher does not pass tests due to random chance (as there are only 2 options it must choose between). Each test consists, on average, of 10 periodically sampled texts throughout the random walk. The test is considered a failure if even one sample fails, ensuring that the distinguisher can accurately deduce the origin throughout the entire attack. Additionally, we find that when the distinguisher fails, it almost always selects response A as the origin. We believe this to be due to positional bias. To combat this, every sample is distinguished as a best-of-2, where the second trial swaps response A and response B. The sample is considered a failure if either trial fails, minimizing the chance that the distinguisher succeeds by guessing. We remark that in every instance where the distinguisher failed, the best-of-2 was a tie, and zero samples (out of 57650) were incorrectly distinguished in both trials. This suggests the model was sometimes unable to overcome positional bias, but was never directly wrong. In these cases, we defer the final verdict to a stronger model, ultimately resulting in 100% accuracy.

The following Python code defines the main method of `SimpleDistinguisher`, implemented with `guidance` (Lundberg et al., 2022) to support constrained decoding. This function is designed to analyze response similarity while leveraging a predefined persona. The persona guides the AI model in determining whether a given perturbed response (`P`) originated from one of two original responses (`A` or `B`).

```
persona = \
"""
You are an expert in analyzing the similarity of responses.
You specialize in detecting whether one response has been derived from another by carefully analyzing the content and structure of the text.
Your expertise in linguistic analysis allows you to distinguish which responses are the most closely related.
Your goal is to provide a clear, concise, and accurate assessment of the provided instructions.
"""

@guidance
def distinguish(lm, persona, **kwargs):
    if persona:
        with system():
            lm += f"{persona}"
    with user():
        lm += f"""
        ### Task Description:
        1. Read the following 3 responses. They are labelled "A", "B", and "P".
        2. "A" and "B" are original responses, and "P" was generated by perturbing one of the original responses.
        3. Identify which of the original responses the perturbed story was derived from.

        Original response A:
        {kwargs["A"]}

        Original response B:
        {kwargs["B"]}

        Perturbed response P:
        {kwargs["P"]}
        """
    with assistant():
        lm += f"""\
        I believe the perturbed response P was derived from original response {select(["A", "B"], name="choice")}.
        """
```

```
    return lm
```

Listing 1: Distinguisher Prompt Template

C.1 A Challenging Distinguisher Example

As an example of the data used with the prompt above, we present an example that was particularly challenging for our distinguishers. After 108 sentence-level perturbations, `GPT-4o` was unable to accurately distinguish the origin. With around 30 seconds of reasoning, `o3-mini-high` correctly distinguished the origin, but had to correct itself while reasoning. We remark that the final section on NASA's Artemis program makes distinguishing this example trivial for humans, suggesting that our distinguishers are significantly weaker than humans. The perturbed text, along with the two original responses, are provided below with some key phrases in bold.

Perturbed Text (GPT-4o Failed to Distinguish)

> Venturing into space is a groundbreaking endeavor that unlocks a multitude of benefits, extending far beyond the realms of scientific discovery and territorial growth. Space exploration, frequently overlooked, is a catalyst for scientific progress, driving the development of pioneering technologies and addressing humanity's most pressing challenges directly, making it a pursuit of paramount importance that warrants greater acknowledgment and support. This essay examines the importance of space exploration, its potential to broaden our understanding, and its ability to contribute to resolving critical global challenges like environmental decay and resource exhaustion. Understanding the cosmos is vital, as it allows us to grasp the intricate mechanisms **governing the universe and our place within it**, ultimately expanding our comprehension of reality itself. Exploring the vastness of the universe reveals a profound comprehension of the fundamental laws that shape reality, the origin of life, and the intricate chronology of cosmic evolution that has spanned eons of time. Delving deeper into our environment not only quenches our innate desire for knowledge but also empowers us to make more informed choices about the planet's destiny, thereby shaping our relationship with the world that surrounds us. The pursuit of space exploration has far-reaching consequences, resulting in numerous groundbreaking discoveries that cumulatively contribute to a significant improvement in global well-being, manifesting in a multitude of tangible advantages. The rapid evolution of technology, encompassing satellite communication, GPS, and medical imaging, has significantly influenced our daily routines, work, and relationships, transforming the way we interact and live our lives. Advances in technology have not only bridged the world but have also led to better health outcomes worldwide, significantly impacting our daily lives and perceptions. Beyond its contributions to science and technology, space exploration provides a **distinctive vantage point for understanding the Earth and its interconnected systems**. Viewing our planet from space offers a comprehensive understanding of the interconnectedness of Earth's atmospheric, oceanic, and terrestrial systems, showcasing a cohesive entity that surpasses its individual components in complexity and unity. Understanding the effects of human actions on the environment is crucial for tackling pressing global issues, such as climate change, which is becoming more apparent with each passing day. Satellite imagery has been instrumental in tracking climate shifts, monitoring the growth of our oceans, and forecasting extreme weather events, all of which are crucial for comprehending the intricate dynamics of our planet's ever-changing environment. Creating a resilient and lasting future demands a sophisticated understanding of climate change's multifaceted impacts and the strategic application of targeted solutions to minimize its effects with accuracy and efficiency. One of the most significant benefits of space exploration is its potential to alleviate the consequences of dwindling resources on Earth. As the global population grows, the need for fundamental resources like food, water, and power is escalating, necessitating a significant boost in production to meet the rising demand. Exploring space can reveal previously unknown sources of essential materials, including minerals and water, which can be exploited to satisfy

the growing need for these vital resources. **By 2024, NASA's Artemis mission seeks to create a self-sustaining lunar community**, ultimately paving the way for a permanent human settlement on the Moon that enables ongoing exploration and habitation. This expedition may reveal new helium-3 deposits, a scarce isotope that could serve as a fuel for nuclear fusion reactions, offering a potential source of power. Ultimately, space exploration is crucial for expanding our understanding of the universe, fostering groundbreaking technological advancements, and tackling pressing global issues such as environmental degradation and resource depletion. Venturing into the unknown reaches of space not only expands our comprehension of the cosmos, but also paves the way for groundbreaking discoveries that, in turn, significantly influence our daily lives and overall well-being. Satellite imaging provides a distinct viewpoint, allowing us to comprehend the intricacies of the obstacles we encounter and tackle them with greater success. As we venture deeper into the cosmos, we may stumble upon innovative answers to humanity's most critical challenges, thereby ensuring a thriving and sustainable world for future generations to inherit.

Original Response A

Space exploration is an essential endeavor for humanity, offering numerous benefits that extend beyond the realm of scientific discovery. While some may view it as a luxury or a frivolous pursuit, the reality is that space exploration plays a vital role in advancing human knowledge, driving technological innovation, and addressing some of the most pressing global challenges we face today. In this essay, we will explore the importance of space exploration, its contributions to human understanding, and its potential to help mitigate global crises like climate change and resource scarcity. One of the primary reasons space exploration is essential is that it allows us to expand our **understanding of the universe and our place within it**. By exploring the cosmos, we gain valuable insights into the fundamental laws of physics, the origins of life, and the evolution of the universe. This knowledge not only satisfies our innate curiosity but also informs our understanding of the world around us, enabling us to make more informed decisions about our planet and its resources. Furthermore, space exploration drives technological innovation, which has numerous spin-off benefits for society. The development of new technologies, such as satellite communications, GPS, and medical imaging, has transformed the way we live, work, and interact with one another. These innovations have also improved our daily lives, from enabling global connectivity to enhancing healthcare outcomes. In addition to its scientific and technological benefits, space exploration also offers a **unique perspective on the Earth and its systems**. By studying the Earth from space, we gain a deeper understanding of the complex interactions between our atmosphere, oceans, and landmasses. This knowledge is essential for addressing global challenges like climate change, where the impacts of human activity on the planet are becoming increasingly evident. Space-based observations have played a critical role in monitoring climate change, tracking sea-level rise, and predicting weather patterns. These insights are essential for developing effective strategies to mitigate the effects of climate change and ensure a sustainable future. Another significant benefit of space exploration is its potential to address resource scarcity. As the global population continues to grow, the demand for resources like food, water, and energy is increasing. Space exploration offers a unique opportunity to identify new sources of resources, such as minerals and water, that can help meet this demand. For example, **NASA's Artemis program aims to return humans to the Moon by 2024**, with the long-term goal of establishing a sustainable presence on the lunar surface. This mission could potentially unlock new resources, such as helium-3, a rare isotope that could be used as fuel for nuclear fusion. In conclusion, space exploration is essential for advancing human knowledge, driving technological innovation, and addressing global challenges like climate change and resource scarcity. By exploring the cosmos, we gain valuable insights into the universe and our place within it, while also developing new technologies that improve our daily lives. Furthermore, space-based observations offer a unique perspective on the Earth and its systems, enabling us to better understand and address the complex challenges we face. As

we continue to explore the vast expanse of space, we may uncover new solutions to some of humanity's most pressing problems, ensuring a sustainable and prosperous future for generations to come.

Original Response B

Space exploration is an essential endeavor for humanity, offering numerous benefits that extend beyond the realm of scientific discovery. It plays a vital role in advancing human knowledge, driving technological innovations, and addressing pressing global challenges such as climate change and resource scarcity. In this essay, we will explore the significance of space exploration and its potential to shape the future of our planet and beyond. The pursuit of space exploration is often viewed as a costly and ambitious endeavor, but it is essential to recognize the significant contributions it makes to our **understanding of the universe and the world we inhabit**. By venturing into space, we gain insights into the fundamental laws of physics, the origins of life, and the evolution of the cosmos. These discoveries not only expand our scientific knowledge but also inspire new generations of scientists, engineers, and innovators. Furthermore, space exploration has led to numerous technological innovations that have transformed various aspects of our daily lives. From the development of GPS and telecommunications to medical imaging and weather forecasting, the spin-off benefits of space exploration have been substantial. These innovations have improved the quality of life for millions of people around the world and have also generated significant economic benefits. In addition to its scientific and technological benefits, space exploration also offers a **unique opportunity to address pressing global challenges**. For instance, the study of Earth from space provides critical insights into the health of our planet and the impacts of climate change. Satellite imaging and remote sensing technologies have enabled scientists to monitor deforestation, track ocean currents, and detect changes in global temperature patterns. This information is essential for developing effective strategies to mitigate the effects of climate change and promote sustainable development. Another significant benefit of space exploration is its potential to provide new resources and opportunities for economic growth. As the world's population continues to grow, the demand for resources such as food, water, and energy will increase. Space exploration offers a way to address this challenge by accessing new sources of resources, such as asteroid mining and lunar helium-3 extraction. These resources could provide a clean and sustainable source of energy, reducing our reliance on fossil fuels and mitigating the impacts of climate change. Finally, space exploration offers a unique opportunity for international cooperation and diplomacy. In an era marked by increasing global tensions and conflict, space exploration provides a shared goal that can bring nations together. Collaborative efforts such as the International Space Station and the **Artemis program** have demonstrated the potential for space exploration to foster global cooperation and understanding. In conclusion, space exploration is essential for advancing human knowledge, driving technological innovations, and addressing pressing global challenges. Its significance extends beyond the realm of scientific discovery, offering numerous benefits that have the potential to shape the future of our planet and beyond. As we continue to explore the vastness of space, we must recognize the importance of investing in this endeavor and working together to address the challenges that lie ahead. By doing so, we can ensure that the benefits of space exploration are shared by all and that the next generation of scientists, engineers, and innovators is inspired to reach for the stars.

D Extended Distinguisher Study

In addition to the main **RQ1** result, we designed an even more challenging evaluation setting to test whether sufficient mixing could obscure the lineage of perturbed texts. Specifically, we focus on the strongest $\mathbf{P}$, `SentenceMutator`, as it previously demonstrated the highest capacity to evade detection by `Llama-3.1-70B`. To amplify its effect, we increase the perturbation budget from 150 to 500 steps, allowing the random walk significantly more opportunities to approach the stationary distribution.

Additionally, we constrain the attack to texts generated from the lowest-entropy prompts, ensuring that candidate parent texts are highly similar to one another. This combination of (1) the strongest perturbation oracle, (2) an extended attack budget, and (3) a highly confounded candidate pool creates the most difficult setting for lineage attribution. If mixing is truly effective under these conditions, we would expect distinguishability to approach random chance.

We find that although the task was more challenging, with more failures on average, `o3-mini-high` still had no issues in distinguishing the origin in each test.

P Oracle	Steps	Tests	`Llama-3.1-70B`	`GPT-4o`	`o3-mini-high`
`Sentence`	500	54	13	3	0
Cumulative Distinguished (%)			**75.9**	**94.4**	**100**

Table 7: Summary of failed distinguisher tests on the most challenging settings. Classification is first performed by `Llama-3.1-70B`, followed by `GPT-4o` on its failures, then `o3-mini-high` on any remaining cases. The overall 100% success rate indicates that the attacked texts never lose memory of their starting points, contradicting **KA1** and suggesting that a stationary distribution is not reached in practice.

D.1 Breakdown by Domain and Entropy

We find domain to be significant in distinguishability, but surprisingly, not entropy.

Domain	Failed Distinguishes (Main)	Failed Distinguishes (Challenge)
Journalism	6/1458	0/18
Creative Writing	7/1560	6/18
Education	40/1537	7/18

Table 8: Domain distribution for tests which `Llama-3.1-70B` failed to distinguish.

Entropy	Failed Distinguishes (Main)	Failed Distinguishes (Challenge)
1	7/462	N/A
2	4/457	N/A
3	8/468	N/A
4	9/462	N/A
5	3/450	N/A
6	1/468	N/A
7	6/450	N/A
8	2/438	N/A
9	5/456	N/A
10	8/444	13/54

Table 9: Entropy distribution for tests which `Llama-3.1-70B` failed to distinguish.

E Appendix: Quality Oracles

E.1 Oracle Details

The quality oracles determine whether the perturbations introduced by various $\mathbf{P}$ preserve the original text's quality. Each oracle operates by querying an LLM with a prompt and some continuation text using different strategies to assess preservation of meaning, fluency, and coherence. The quality decision is based on whether the mutated text is judged to be as good as or better than the original. All oracle queries include the original prompt to provide context for evaluation.

We implement and evaluate eight distinct quality oracles using `guidance` (Lundberg et al., 2022) to support constrained decoding for ranking, scoring, and preference based assessments.

- **`RankOracle`**: $\mathbf{Q}$ is prompted to rank the two responses in terms of preference, and the order of texts is then reversed in a second query. If the mutated text is preferred in both cases, quality is considered preserved.
- **`SoloOracle`**: $\mathbf{Q}$ is prompted twice, independently grading each text on a numerical scale. If the mutated text receives a score equal to or higher than the original, its quality is considered preserved.
- **`JointOracle`**: Similar to `SoloOracle`, but $\mathbf{Q}$ assigns numerical scores to both texts in the same prompt. The order is flipped in a second query. If the mutated text scores equal to or higher in both cases, quality is preserved.
- **`RelativeOracle`**: $\mathbf{Q}$ is prompted to select the better response or declare a tie, repeating the query with the order reversed. If the mutated text is chosen in both cases or a tie is declared, quality is preserved.
- **`BinaryOracle`**: $\mathbf{Q}$ is asked a direct yes/no question: "Is the mutated text just as good or better than the original?" If the response is "yes", quality is preserved.
- **`MutationOracle`**: Similar to `BinaryOracle`, but the prompt explicitly states that the mutated text is a modification of the original. The query is repeated with the order reversed. If both responses are "yes", quality is preserved.
- **`ExampleOracle`**: Similar to `BinaryOracle`, but includes an example (1-shot prompting) before presenting the actual texts. If the response is "yes", quality is preserved.
- **`DiffOracle`**: $\mathbf{Q}$ is provided with the original text, mutated text, and a computed diff between them. It is asked whether these changes are acceptable. If the response is "yes", quality is preserved.

These oracles serve as key components in our evaluation framework, allowing us to systematically assess how to best approximate human judgments of quality. By incorporating multiple prompting strategies, we ensure robustness in our analysis of watermark perturbation effectiveness.

E.2 Sandcastle Dataset Statistics

Since absolute quality scoring is difficult for humans (Chiang et al., 2024), we formulated the annotation task as pairwise preference judgments with a tie option. Several coauthors, following standardized guidelines, compared perturbed texts to their originals, unaware of which was which. Table 10 shows the class distribution, where we merged "Attacked Better" and "Tie" into a Quality Preserved (QP) category to support binary classification.

	Quality Preserved		**Quality Degraded**	
Split	**Attacked Better**	**Tie**	**Original Better**	**Total**
Train	12	238	306	556
Test	1	103	135	239
Total	13	341	441	795

Table 10: Distribution of human quality assessments by split for the Sandcastles dataset. The table details counts for cases where attacked outputs were rated as "Attacked Better" or "Tie" (grouped under Quality Preserved (QP)) versus "Original Better", along with overall totals for both training and test sets.

E.3 Full Oracle Results

Table 11 provides a detailed comparison of quality oracles, including inference time, QP Precision, Overall F1, and RewardBench scores where available. Despite fine-tuning, no oracle fully aligns with human judgments, and high RewardBench scores do not guarantee strong performance in our setting. Proprietary models like `GPT-4o` with fine-tuning perform best but are impractical for large-scale attacks. Locally hosted models (`MutationOracle`, `DiffOracle`) offer a viable alternative but still misclassify degraded outputs. These results highlight the challenges of using LLM-based oracles for reliable watermark attack guidance.

Oracle	Model	Type	Time (s)	QP Prec.	Overall F1	RB Score
SkyworkOracle	Skywork-Reward-Gemma-2-27B-v0.2 (Liu et al., 2024c)	FLOAT	2.22	43.51	26.39	94.3
RankOracle	Llama-3.1-70B-Instruct (Dubey et al., 2024)	BOOL	4.33	50.00	37.09	–
SoloOracle	Llama-3.1-70B-Instruct (Dubey et al., 2024)	INT	2.23	49.49	39.86	–
JointOracle	Llama-3.1-70B-Instruct (Dubey et al., 2024)	INT	3.62	53.85	40.85	–
INFORMOracle	INF-ORM-Llama3.1-70B (Minghao, 2024)	FLOAT	5.81	65.63	54.40	95.1
QRMOracle	QRM-Gemma-2-27B (Dorka, 2024)	FLOAT	3.28	50.68	56.98	94.4
RelativeOracle	Llama-3.1-70B-Instruct (Dubey et al., 2024)	CLASS	2.76	79.59	63.07	–
ExampleOracle	Llama-3.1-70B-Instruct (Dubey et al., 2024)	BOOL	1.33	79.59	63.07	–
BinaryOracle	Llama-3.1-70B-Instruct (Dubey et al., 2024)	BOOL	1.27	61.90	63.82	–
ArmoRMOracle	ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024)	FLOAT	0.33	65.71	64.26	90.4
OffsetBiasOracle	Llama-3-OffsetBias-RM-8B (Park et al., 2024)	FLOAT	0.32	62.22	65.30	89.6
Prometheus2Absolute	prometheus-8x7b-v2.0 (Kim et al., 2024)	FLOAT	7.28	74.78	66.73	74.5
Prometheus2Relative	prometheus-8x7b-v2.0 (Kim et al., 2024)	BOOL	7.36	74.78	66.73	74.5
Prometheus2Absolute	GPT-4o (OpenAI, 2024b)	INT	7.93	76.70	66.87	–
MutationOracle	Llama-3.1-70B-Instruct (Dubey et al., 2024)	BOOL	2.74	**84.62**	66.93	–
Prometheus2Relative	GPT-4o (OpenAI, 2024b)	BOOL	7.73	77.23	67.05	–
Prometheus2Relative	GPT-4-Turbo (OpenAI, 2024a)	BOOL	11.94	75.00	67.27	–
Prometheus2Absolute	GPT-4-Turbo (OpenAI, 2024a)	INT	12.46	76.15	67.55	–
InternLMOracle	internlm2-20b-reward (Cai et al., 2024)	FLOAT	0.86	65.69	69.84	90.6
DiffOracle	Llama-3.1-70B-Instruct (Dubey et al., 2024)	BOOL	1.83	71.74	70.85	–
MutationOracle+FT	Llama-3.1-70B-Instruct + Fine-tuning	BOOL	3.25	81.18	71.83	–
DiffOracle+FT	Llama-3.1-70B-Instruct + Fine-tuning	BOOL	1.80	69.07	76.94	–
DiffOracle+FT	GPT-4o (OpenAI, 2024b) + Fine-tuning	BOOL	0.46	75.51	77.32	–
MutationOracle+FT	GPT-4o (OpenAI, 2024b) + Fine-tuning	BOOL	0.84	74.51	**77.38**	–

Table 11: Overview of oracle performance on our human-annotated test set. For each oracle we report average inference time, Quality-Preserved (QP) Precision, Overall F1, and RewardBench (RB) Score when available. Despite fine-tuning on human judgements, no oracle perfectly capture human quality assessments, and high RB Scores did not predict strong performance in our evaluation setting.

E.4 `InternLM` vs `DiffOracle`

We compared the proportion of cases where humans agreed that the quality of generated outputs was preserved. The results, summarized in Table 12, show that `InternLMOracle` had a higher agreement rate (47.78%) than `DiffOracle+FT` (40.0%).

Oracle	Agree QP	Disagree QP
DiffOracle	40.00	60.00
InternLM	47.78	52.22

Table 12: Comparison of human agreement rates on quality preservation (QP) percentages between DiffOracle and InternLM. InternLM shows a higher agreement rate, suggesting it aligns better with human judgments.

To quantify the probability that `InternLMOracle` is genuinely the better oracle, we adopt a Bayesian approach, modeling the probability of agreement for each oracle as a Beta distribution:

$$p_A \sim \text{Beta}(A+1, N_A - A + 1),$$
$$p_B \sim \text{Beta}(B+1, N_B - B + 1)$$

where A and B are the counts of human agreement for `DiffOracle+FT` and `InternLMOracle`, respectively, and N_A and N_B are the total evaluations for each oracle.

Using a Monte Carlo simulation with 100,000 samples, we estimate:

$$P(p_B > p_A) \approx 85.08\%$$

indicating that `InternLMOracle` has an 85.08% probability of being the better judge in preserving quality according to human evaluators. Given this high confidence, we justify the use of `InternLMOracle` as the preferred oracle for further evaluations[5].

F Appendix: Human Annotation Details

Annotators were provided with the following instructions when reviewing:

- Determine which is a better response to the prompt: text A, text B, or tie.
- Judge quality based on content, style, cohesion, and prompt relevance.
- Note: Formatting is not especially important for quality (e.g. paragraph breaks should be ignored).

These guidelines ensured that evaluations focused on meaningful quality differences rather than superficial formatting artifacts.

[5]This surprising reversal of performance may be attributable to `DiffOracle+FT` managing too much noise in the changelog of edits when attacks exceed 20 steps (the maximum attack length present in the Sandcastles dataset).

G Appendix: Extended Attack Results Analysis

Table 13 provides a detailed breakdown of attack performance, including automated quality metrics, revealing several notable patterns. One interesting finding is that, in some cases, attacks appear to "improve" certain quality metrics, such as perplexity and grammar error rates. This effect is most pronounced for the Adaptive watermark, where the average perplexity and grammar errors decrease post-attack. However, this improvement is largely driven by a few low-quality outliers in the original watermarked dataset, rather than a systematic enhancement of text fluency. Despite these reductions in surface-level errors, the `InternLM` quality score consistently drops, indicating that attacks tend to reduce overall coherence and relevance, even when fluency-related metrics superficially improve.

Another trend is that unique bigram diversity (μ_{d_t}) increases slightly in many cases, particularly for sentence- and document-level attacks. This suggests that perturbations introduce more varied word sequences, potentially disrupting structured patterns imposed by watermarking. However, this increase is relatively small, meaning that while attacks may inject lexical diversity, they do not necessarily enhance the text in a meaningful way. Instead, the most aggressive perturbation strategies—particularly sentence- and document-level attacks—cause the largest drops in the `InternLM` quality score, reinforcing the idea that these attacks are the most disruptive to text coherence. While they achieve the highest watermark removal rates, they also tend to introduce noticeable degradation, making the resulting text less natural and readable.

By contrast, perturbation strategies that fail to effectively break watermarks, such as word-level and entropy-based edits, also have minimal impact on quality metrics. This suggests that these finer-grained mutations are too minor to erase watermark signals while also being too weak to meaningfully degrade text fluency. More broadly, the average attack success rate remains relatively low even before enforcing quality constraints, with ASR_{fin} at only 26.13%. After accounting for quality degradation, this drops further to just 10.47%, confirming that successfully removing watermarks without compromising text quality remains a substantial challenge for adversaries.

Watermark	P Oracle	μ_{w_0}	μ_{w_t}	BP	ASR_{min}	ASR_{fin}	Reviewed	QP	¬QP	$\text{Q-ASR}_{\text{fin}}$	μ_{q_0}	μ_{q_t}	μ_{p_0}	μ_{p_t}	μ_{g_0}	μ_{g_t}	μ_{d_0}	μ_{d_t}
Adaptive	Word	99.27	70.37	56.16	0.00	0.00	0	0	0	0.00	0.45	0.01	63.32	77.64	16.21	13.79	603.64	609.09
Adaptive	EntropyWord	99.27	82.45	56.16	0.00	0.00	0	0	0	0.00	0.45	-0.05	63.32	78.01	16.21	19.00	603.64	608.33
Adaptive	Span	99.27	67.21	56.16	1.54	1.54	2	2	0	1.54	0.45	0.06	63.32	44.34	16.21	11.24	603.64	616.06
Adaptive	Sentence	99.27	59.93	56.16	35.34	19.21	20	8	12	7.68	0.45	0.24	63.32	27.78	16.21	5.42	603.64	602.26
Adaptive	Document	99.27	58.55	56.16	48.78	45.24	20	8	12	18.10	0.45	0.00	63.32	16.27	16.21	2.11	603.64	464.31
Adaptive	Document1Step	99.27	70.94	56.16	1.16	1.16	2	2	0	1.16	0.45	0.32	63.32	30.09	16.21	0.46	603.64	541.44
Adaptive	Document2Step	99.27	73.39	56.16	5.33	4.71	8	5	3	2.94	0.45	0.27	63.32	27.48	16.21	8.06	603.64	580.33
KGW	Word	0.28	0.17	0.21	47.54	20.00	20	4	16	4.00	0.16	-0.30	8.87	30.30	4.24	11.10	479.90	572.57
KGW	EntropyWord	0.28	0.22	0.21	3.45	0.56	1	0	1	0.00	0.16	-0.31	8.87	20.15	4.24	10.14	479.90	535.14
KGW	Span	0.28	0.20	0.21	38.46	32.35	20	14	6	22.65	0.16	-0.29	8.87	14.78	4.24	7.19	479.90	536.79
KGW	Sentence	0.28	0.14	0.21	89.47	56.52	20	7	13	19.78	0.16	0.02	8.87	16.93	4.24	4.68	479.90	658.71
KGW	Document	0.28	0.18	0.21	62.50	44.44	20	8	12	17.78	0.16	-0.27	8.87	10.48	4.24	7.70	479.90	435.39
KGW	Document1Step	0.28	0.27	0.21	12.66	8.54	14	7	7	4.27	0.16	0.07	8.87	9.47	4.24	0.31	479.90	482.51
KGW	Document2Step	0.28	0.18	0.21	9.09	7.78	10	4	6	3.11	0.16	-0.03	8.87	11.97	4.24	4.41	479.90	501.89
SIR	Word	5.32	1.74	1.27	78.22	57.89	20	1	19	2.89	0.43	0.00	9.56	29.96	2.86	6.79	512.44	588.33
SIR	EntropyWord	5.32	3.30	1.27	39.68	27.54	20	0	20	0.00	0.43	-0.03	9.56	20.20	2.86	1.44	512.44	554.38
SIR	Span	5.32	1.57	1.27	60.71	37.40	20	5	15	9.35	0.43	0.03	9.56	15.28	2.86	4.36	512.44	561.04
SIR	Sentence	5.32	0.52	1.27	87.65	74.71	20	13	7	48.56	0.43	0.30	9.56	17.97	2.86	2.93	512.44	597.44
SIR	Document	5.32	0.93	1.27	61.54	46.09	20	6	14	13.83	0.43	0.10	9.56	12.65	2.86	10.64	512.44	454.21
SIR	Document1Step	5.32	2.54	1.27	14.04	14.04	12	11	1	12.87	0.43	0.23	9.56	11.74	2.86	0.26	512.44	491.45
SIR	Document2Step	5.32	3.07	1.27	68.09	49.06	20	12	8	29.44	0.43	0.26	9.56	11.70	2.86	4.29	512.44	530.79
Averages					36.44	26.13		40.48	59.52	10.47	0.35	0.03	27.25	25.49	7.77	6.49	532.00	548.69

Table 13: Attack success rates (ASR) and autoamated quality scores across different perturbation strategies. Human review reveals an average of 59.52% of successfully attacked texts have degraded quality. $\boldsymbol{\mu_{w_0}}$ represents the initial watermark score at step 0, while $\boldsymbol{\mu_{w_t}}$ represents the final watermark score after t mutation steps.“min” refers to the point where the watermark score is at its lowest during the attack while “fin” refers to score at the final step of the attack. "Reviewed" indicates the number of human-reviewed examples where the watermark was broken. **QP** and **¬QP** represent the number of cases where human reviewers judged the attacked text as quality-preserving or degraded, respectively. $\textbf{Q-ASR}_{\text{fin}}$ is the re-estimated attack success after controling for quality, calculated as $\textbf{ASR}_{\text{fin}} \times (\text{QP}/\text{Reviewed})$. The remaining quality columns show, for time step 0 and the final step t, `InternLM` quality score (q), perplexity (p), grammar errors (g), and unique bigram diversity (d). On average, quality degraded significantly while perplexity, grammar errors, and diversity improved.

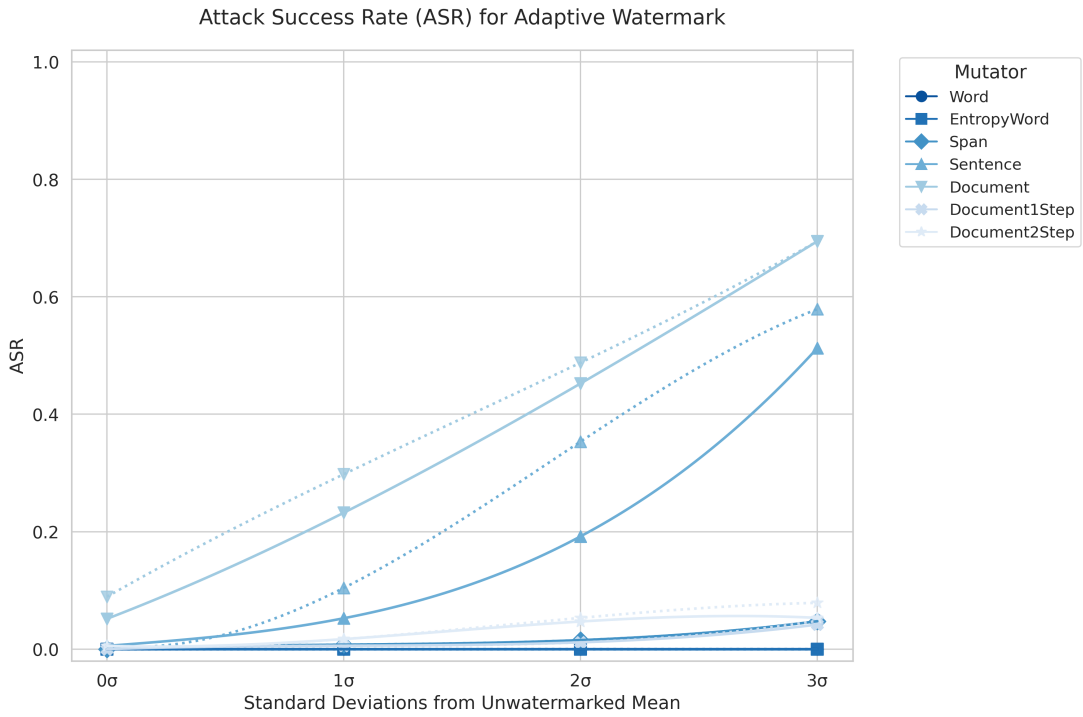

Figure 5: Attack success rate (ASR) vs. detection threshold for the Adaptive watermarking scheme. Each curve represents a different perturbation oracle, with thresholds measured in standard deviations above the unwatermarked mean.

G.1 Attack Success Rate vs. Detection Threshold

Figures 7, 6, and 5 plot the final attack success rate (**ASR**) for each perturbation oracle under the **KGW**, **SIR**, and **Adaptive** watermarking schemes, respectively. The horizontal axis represents detection thresholds measured in standard deviations above the mean detection score for unwatermarked text (i.e., 0σ, 1σ, 2σ, or 3σ). A higher threshold allows more texts to be considered "unwatermarked," so ASR generally increases as we move to the right. The vertical axis indicates the fraction of attacked texts that fall below each threshold once all permitted mutations have been applied.

Each curve corresponds to a specific mutator—`Word`, `EntropyWord`, `Span`, `Sentence`, `Document`, `Document1Step`, or `Document2Step`—with line style distinguishing s_{min} (dotted) from s_{fin} (solid). In general, token-level $\mathbf{P}$ (`Word`, `EntropyWord`, `Span`) make smaller, more localized edits, while document-level $\mathbf{P}$ (`Document`, `Document1Step`, `Document2Step`) can restructure larger portions of text. Comparing these curves reveals which $\mathbf{P}$ achieve higher ASR for each watermarking scheme and how sensitive those results are to stricter or looser detection thresholds.

Overall, two main patterns emerge. First, as the detection threshold increases, more perturbed texts evade being flagged, causing the ASR curves to rise. Second, the extent of this rise varies across both watermarking schemes and $\mathbf{P}$: some methods prove more effective at evading detection for KGW or SIR, whereas Adaptive typically shows lower ASR across thresholds. This aligns with our broader observation that larger, more context-aware edits (`Document`-based $\mathbf{P}$) often outperform smaller, token-level edits, but still rarely achieve high success rates without risking noticeable quality degradation.

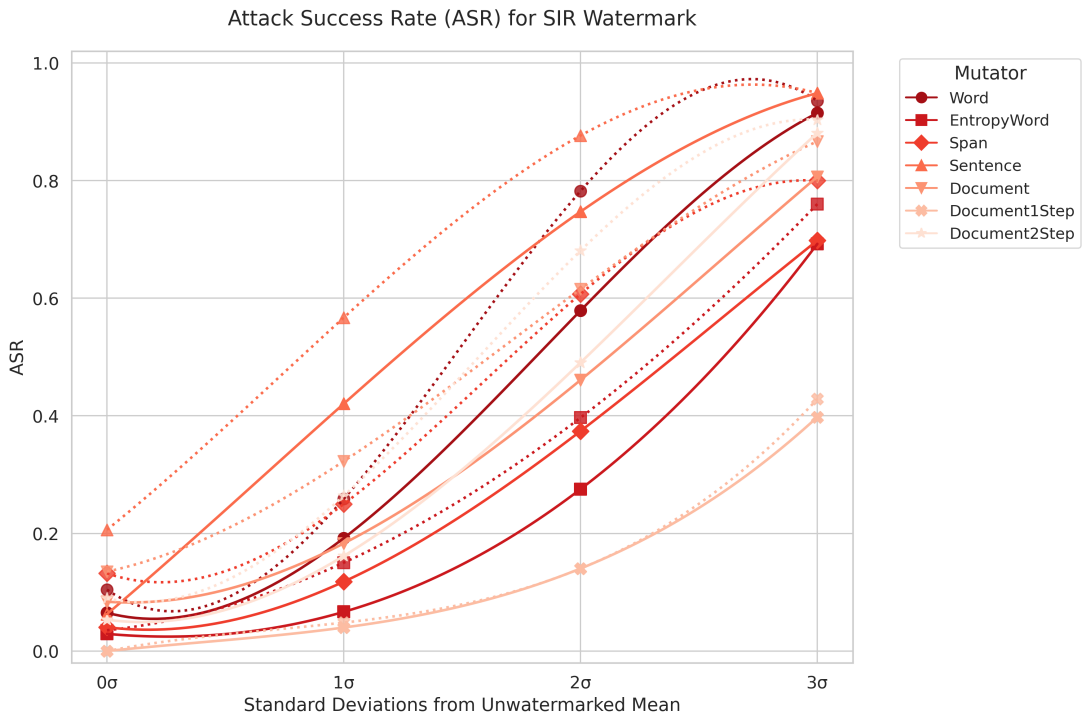

Figure 6: Attack success rate (ASR) vs. detection threshold for the SIR watermarking scheme. The plot shows the fraction of attacked texts falling below various thresholds (in standard deviations above the unwatermarked mean) for multiple perturbation oracles.

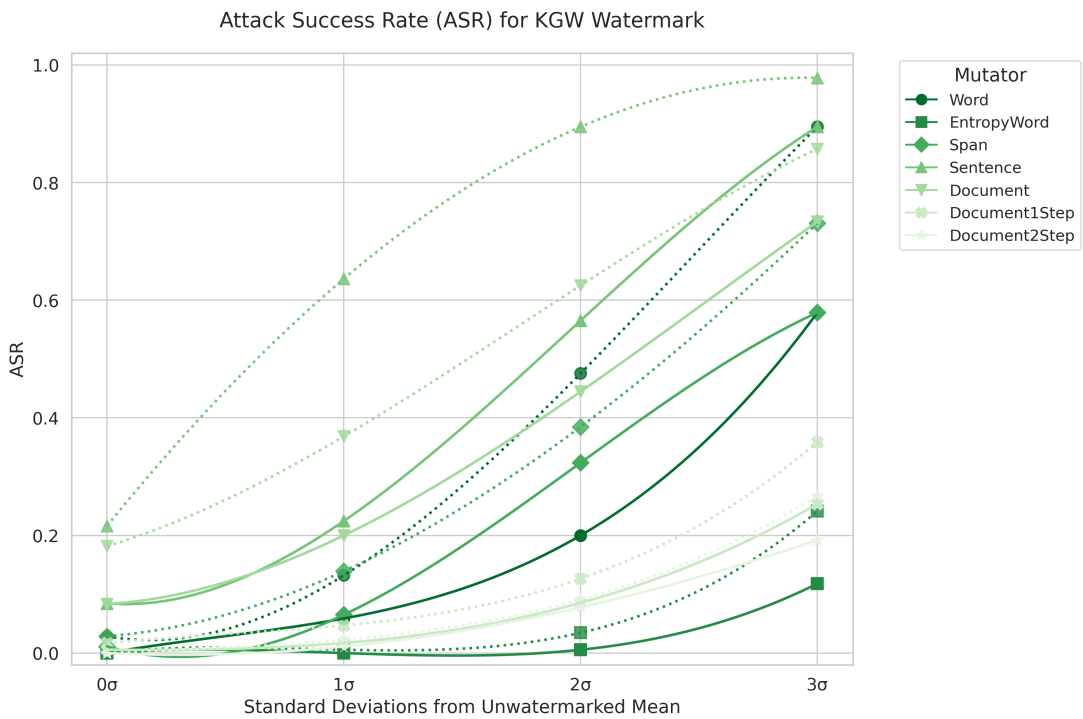

Figure 7: Attack success rate (ASR) vs. detection threshold for the KGW watermarking scheme. Different curves correspond to various perturbation oracles, with the detection threshold defined as standard deviations above the mean detection score of unwatermarked texts.

H Appendix: What factors contributed to attack inefficiency?

The efficiency of the WITS attack against private watermarking schemes is hampered by two interrelated challenges. First, the attack relies on a random walk that must approach its stationary distribution, with the mixing time critically dependent on the second-largest eigenvalue, g, of the transition matrix $\vec{P}$. Not only is computing g exactly infeasible, but even approximating it is extremely difficult. In practice, the size and complexity of $\vec{P}$—which depends on factors such as the mutator, prompt, and quality barrier—make computing any information about $\vec{P}$ computationally intractable. As a result, the attacker must rely on upper bounds for g to estimate the mixing time, a strategy that introduces significant uncertainty into the overall attack duration. Notice that this isn't an issue for public watermarking schemes since the attacker can stop as soon as the watermark is removed.

Second, attempts to accelerate the mixing process—such as by increasing the step size of the perturbation oracle—risk degrading the quality of the text. As quality decreases, so does the success rate of mutations (i.e., the effective constant ϵ_{pert} no longer holds), which in turn negates the benefits of improved mixing by requiring even more iterations to produce acceptable outputs.

In essence, there is a fundamental tension between reducing the mixing time to achieve attack efficiency and maintaining the quality of the attacked text. A more refined theoretical analysis that balances these competing factors is necessary to fully understand the capabilities of the WITS attack. We leave this compelling direction for future work.

Figures 8, 9, 10, and 11 below illustrate the rolling success rate of mutations across various watermarking schemes and mutator types, thereby supporting our first claim. In these computations, the window size is defined as one-tenth of the total number of mutator steps (e.g., for the Sentence Mutator, $150/10 = 15$ steps).

Notably, $\mathbf{P}$ characterized by larger step sizes exhibit lower success rates. Furthermore, the plots reveal a modest correlation between the mutation success rate and the entropy level: prompts with lower entropy tend to have reduced success rates. This phenomenon may be attributable to the fact that lower-entropy prompts are generally longer, thereby increasing the difficulty of generating a mutated response that maintains high quality. Consequently, any interpretation of this correlation should be approached with caution.

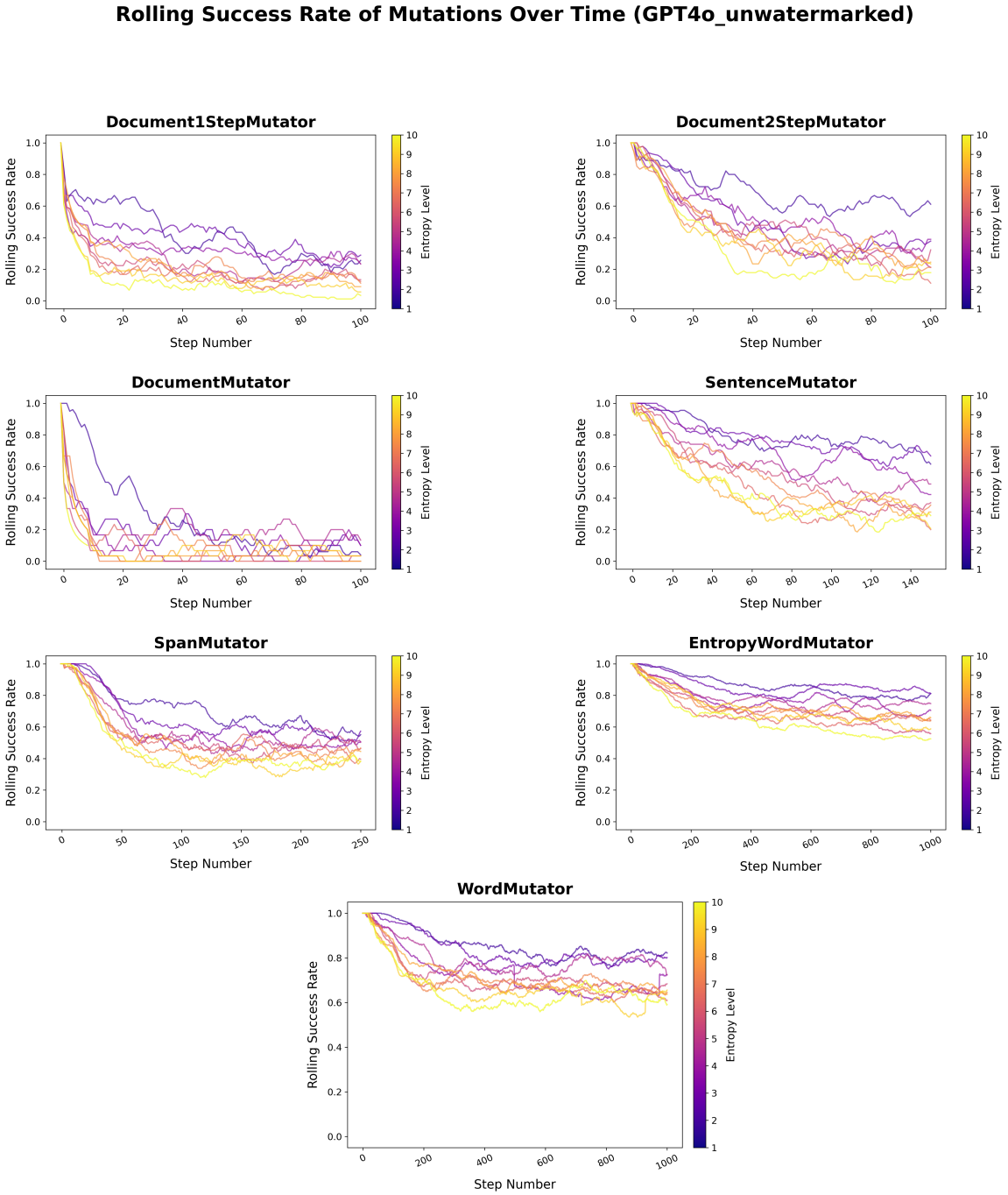

Figure 8: Rolling success rate for GPT-4o generations, which are unwatermarked.

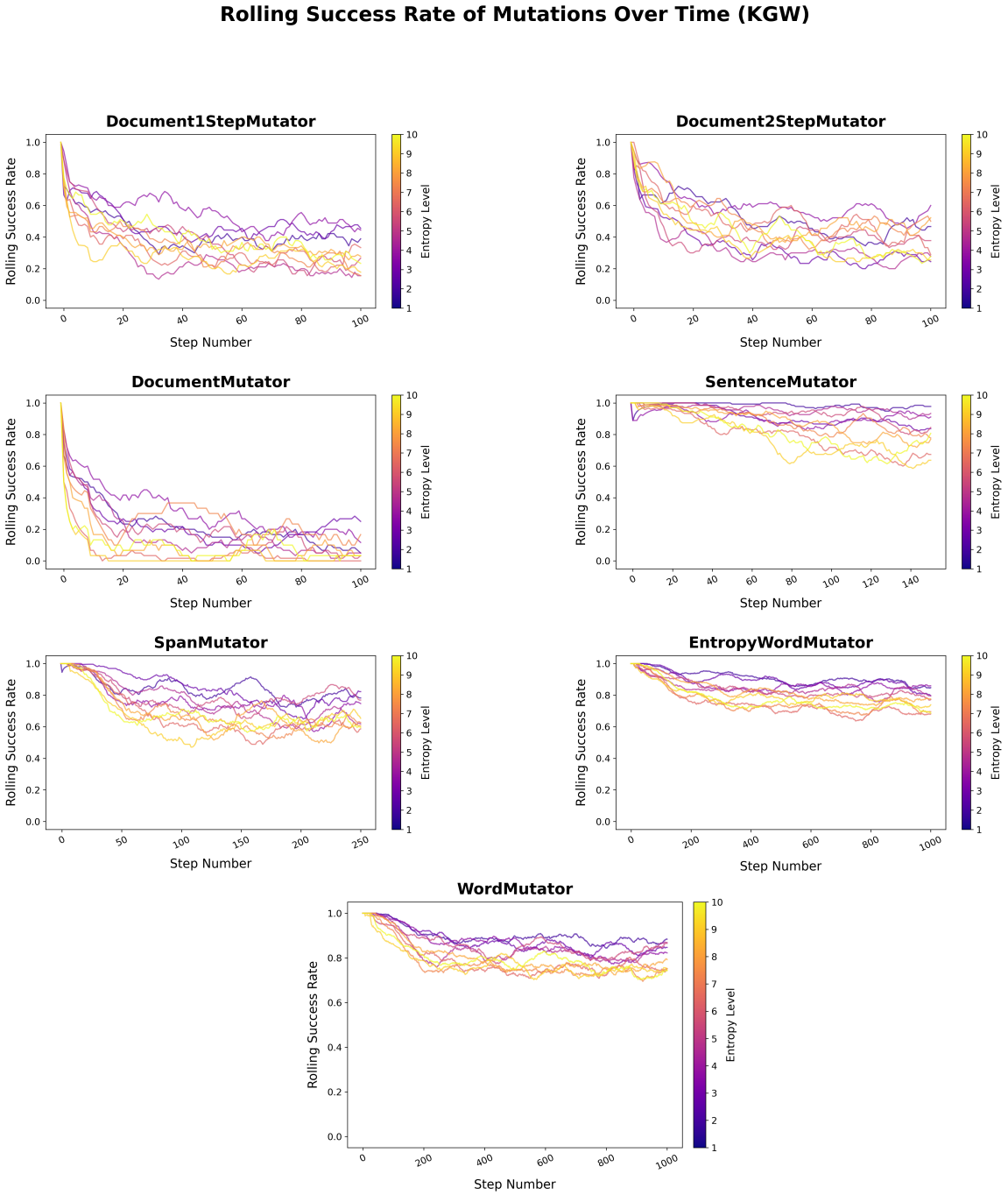

Figure 9: Rolling success rate for the KGW watermark.

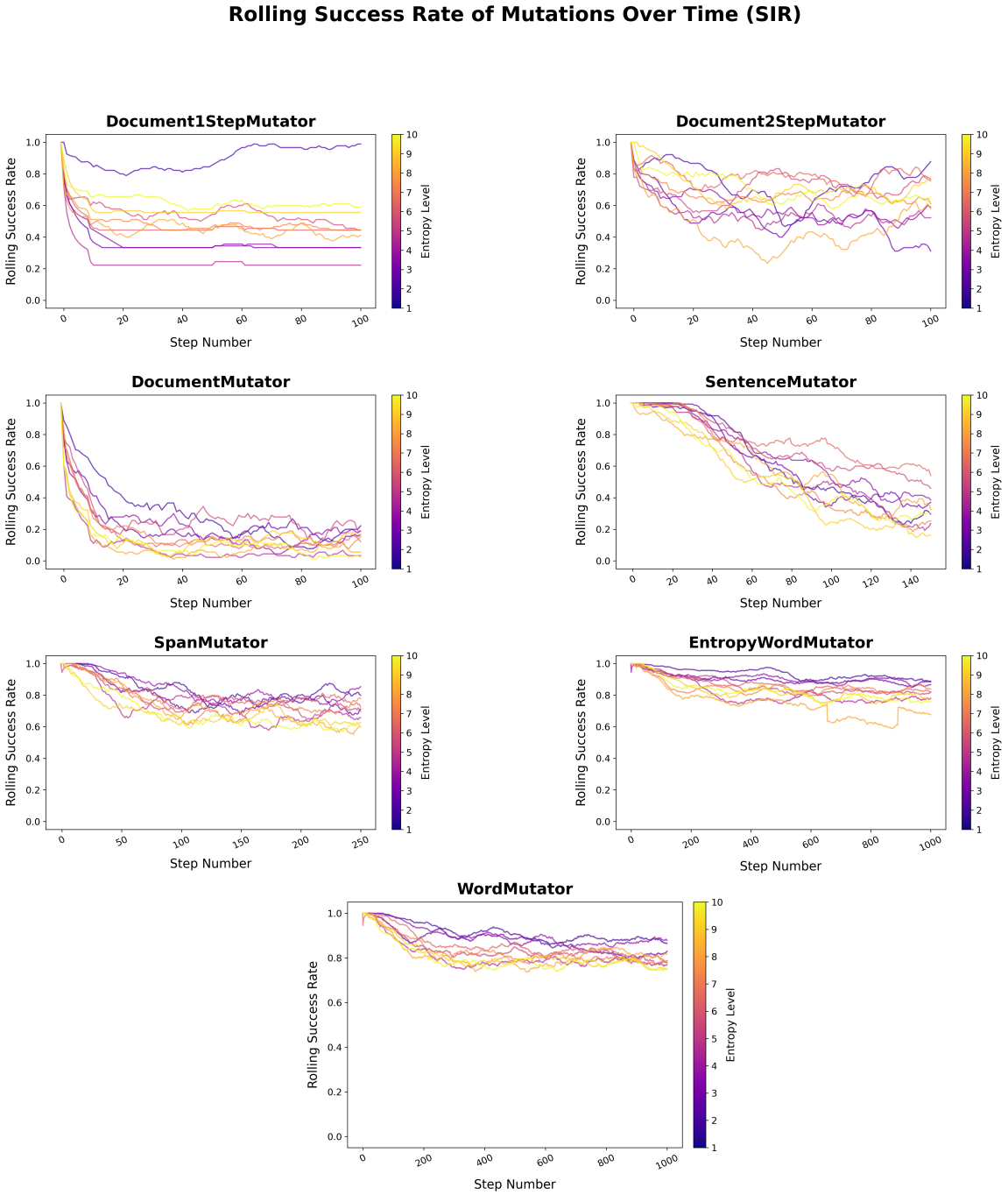

Figure 10: Rolling success rate for the SIR watermark.

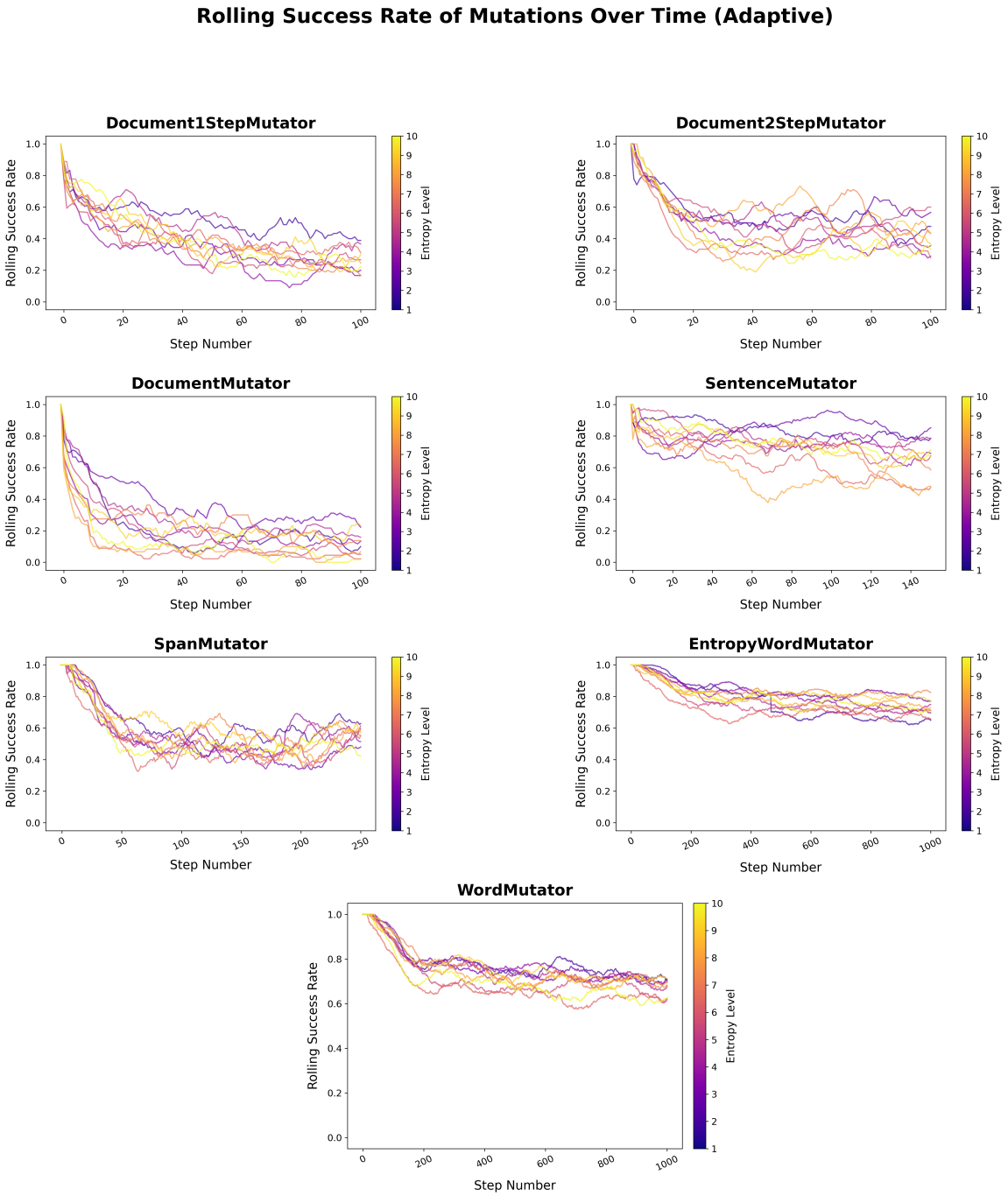

Figure 11: Rolling success rate for the Adaptive watermark.