ORIGINMARK: MITIGATING ATTACKS IN DISTORTION-FREE WATERMARKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Distortion-free watermarking has emerged as a promising approach for establishing the provenance of large language model (LLM) outputs by preserving the original next-token distribution when averaged over watermark randomness. However, this guarantee does not directly characterize practical deployments, where a fixed secret key may be reused across many generation requests. We show that conditioning on a fixed key induces persistent, localized token-level distortions that form an observable watermark fingerprint. Exploiting this structure, we develop two black-box attacks requiring only sampled model outputs: a watermark identification attack that detects context-dependent watermarking and estimates its context width through controlled prefix perturbations, and a key-free spoofing attack that transfers inferred watermark-induced token preferences to a proxy language model without access to model logits, watermark keys, or detector feedback. Across seven open-weight models and three distortion-free watermarking schemes, these attacks expose substantial vulnerabilities under fixed-key deployment; we further observe a response pattern in two publicly accessible Gemini models consistent with context-dependent watermarking. To mitigate these vulnerabilities, we formalize randomized distortion-freeness, which requires the distribution averaged over a deployed key pool to exactly recover the original model distribution at every context, and introduce OriginMark, a watermarking scheme that realizes this property using only eight complementary keys. Experiments show that OriginMark reduces identification and spoofing signals to levels comparable to unwatermarked generation while preserving output quality and diversity. At a 1% false-positive rate, OriginMark achieves a 77.3% detection rate at 400 tokens, outperforming the strongest evaluated attack-resistant baseline by 11.3%.

1 INTRODUCTION

As large language models become capable of generating text that is difficult to distinguish from human-authored content, reliable mechanisms for establishing content provenance have become increasingly important. Watermarking (Aaronson, 2023; Kirchenbauer et al., 2023) provides one approach to this problem by embedding a key-dependent statistical signal during generation that can subsequently be verified by a detector possessing the corresponding secret key. Among existing approaches, *distortion-free watermarking* (Kuditipudi et al., 2024; Christ et al., 2024; Hu et al., 2024; Wu et al., 2024) is particularly appealing because it modifies the next-token distribution in a manner that, when averaged over the watermark key, exactly recovers the original model distribution.

The distortion-free guarantee of modern text watermarks is inherently an average-case statement over watermark keys, yet practical systems typically reuse a fixed secret key across many generations. Under a fixed key, the watermark induces a deterministic, context-dependent distribution that generally differs from the original language-model distribution and, by concavity of Shannon entropy, is more concentrated on average (Chen et al., 2026). We find that this distortion is highly localized: across seven open-weight language models and three state-of-the-art watermarking schemes, only 0.63–2.67 of the model's top-20 candidate tokens receive increased probability on average. Because repeated contexts induce the same context-specific key, these token-level preferences persist across generations, creating a stable and potentially exploitable *watermark fingerprint*.

We show that this fingerprint enables two black-box attacks using only sampled outputs. First, we introduce a *watermark identification attack* that detects context-dependent watermarking and esti-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

mates its context width. The attack perturbs preceding tokens and measures whether the model's preference between candidate next tokens is preserved. Our experiments also reveal response patterns in two Gemini models consistent with SynthID-Text. Second, we develop a key-free *spoofing attack* that infers watermark-favored next-token candidates from queries to the target model and transfers these preferences to a smaller proxy model. Using Qwen3.5-2B to attack a watermarked Qwen3.5-9B model, we achieve nearly 100% watermark acceptance at a 1% false-positive rate.

To mitigate these vulnerabilities, we propose *randomized distortion-freeness*, requiring that, at every context, the watermarked distributions averaged over a randomized deployed key pool exactly recover the original language-model distribution. This formulation preserves distributional fidelity at the deployment level while enabling key randomization as a defense against persistent watermark fingerprints. Based on this principle, we introduce **OriginMark**, which partitions the vocabulary into K channels and associates each key with a distinct promoted channel at every context. The resulting key-conditioned perturbations are constructed to cancel exactly when averaged over the key pool, while retaining the statistical structure required for watermark detection. Empirically, OriginMark reduces both the identification and spoofing attacks to levels comparable to unwatermarked generation, preserves output diversity, and exceeds the strongest evaluated attack-resistant baseline by 11.3 percentage points in true-positive rate at a 1% false-positive rate. Additionally, randomized distortion-freeness naturally satisfies conventional distortion-free requirements, ensuring that our method preserves output quality. Our contributions are summarized as follows:

- We identify a distinction between key-averaged distortion-freeness and fixed-key deployment, showing that repeated use of a single secret key induces persistent token-level preferences that can reduce output diversity and expose an observable watermark fingerprint.
- We develop two attacks that exploit this fingerprint using only sampled outputs, without access to model logits, watermark keys, or detector feedback, and evaluate them on both open-weight models and publicly accessible LLM APIs.
- We formalize randomized distortion-freeness as a pool-wise distributional guarantee and introduce OriginMark, which realizes this property with a small complementary key pool, substantially mitigating the evaluated attacks while preserving output quality, output diversity and maintaining strong detection performance.

2 Related Work

2.1 Distortion-free watermarking

Early text watermarks introduce detectable signals by biasing token selection, while distortion-free schemes preserve the original next-token distribution in expectation over watermark randomness. Recent methods, including SynthID-Text (Dathathri et al., 2024), MCMark (Chen et al., 2025b), and TextSeal (Sander et al., 2026), improve detection and address output diversity. However, distortion-freeness does not guarantee the absence of persistent token preferences under fixed-key deployment. Our work investigates how these preferences enable attacks and introduces complementary-key randomization to eliminate aggregate next-token distortion across a small deployed key pool.

2.2 Watermark identification, spoofing, and defenses

Prior work has developed output-only watermark identification methods (Gloaguen et al., 2025), random selection probing (Chen et al., 2025a), watermark stealing (Jovanovic et al., 2024), and learning-based spoofing techniques such as DITTO (An et al., 2026). Other studies (Christ et al., 2024; Aremu et al., 2025) investigate key randomization and cryptographic undetectability. In contrast, our attacks exploit context-dependent token preferences to identify watermark deployment and spoof watermark signals using only sampled outputs, without recovering secret keys or training a student model. OriginMark addresses the underlying deployment-level vulnerability through exact pool-wise distribution preservation. Further discussion is provided in Appendix C.

3 Attacks on Distortion-Free Watermarks

3.1 Preliminaries: Distortion-Free Watermarks

Let $\mathcal{V}$ denote the vocabulary of a language model M. Given a prompt and a previously generated prefix $x_{<t} = (x_1, \ldots, x_{t-1})$, the model induces a next-token distribution $P_M(x_t \mid x_{<t}), x_t \in$

$\mathcal{V}$. A watermark modifies the token-sampling procedure using the secret key $sk \in \mathcal{K}_0$. At each generation step t, existing context-dependent watermarking schemes (Hu et al., 2024; Dathathri et al., 2024; Sander et al., 2026) typically construct a watermark context $c_t = x_{t-h:t-1} \in \mathcal{V}^h$, where h denotes the *context width*. The context is then combined with the secret key through a pseudorandom function or hash function $H : \mathcal{K}_0 \times \mathcal{V}^h \to \mathcal{K}$, yielding a context-specific watermark key $k_t = H(sk, c_t) \in \mathcal{K}$. Given k_t, the watermark transforms the original next-token distribution through a watermarking function F:

$$P_W(\cdot \mid x_{<t}, k_t) = F\left(P_M(\cdot \mid x_{<t}), k_t\right). \tag{1}$$

Here, P_W denotes the watermarked distribution. A watermark is *distortion-free* if averaging over the watermark randomness exactly recovers the original model distribution. Formally, for every prefix $x_{<t}$ and every token $x_t \in \mathcal{V}$,

$$\mathbb{E}_{k\sim\mathcal{K}}\left[P_W(x_t \mid x_{<t}, k)\right] = P_M(x_t \mid x_{<t}). \tag{2}$$

Therefore, when the watermark key is averaged over its prescribed distribution, watermarking introduces no marginal distortion to the model's next-token probabilities.

3.2 Distortion in Distortion-Free Watermarks

As shown in Eq. 2, distortion-freeness guarantees that the original next-token distribution is recovered only after averaging over the watermark key or watermark randomness. Importantly, this does not imply that the watermarked distribution conditioned on a particular key is identical to the original model distribution. At generation step t, the context-specific key $k_t = H(sk, c_t)$ depends on the previously generated tokens. Meanwhile, the current watermark perturbation influences the sampled token x_t, which subsequently becomes part of the watermark context used at future steps. Therefore, watermark perturbations do not act independently across generation steps. An early watermark-induced token choice changes the future context, which in turn changes the context-specific keys and subsequent perturbations. Under repeated use of the same master key, this cascading interaction may induce persistent, key-specific regularities in generated text. We refer to such regularities as a *watermark fingerprint*. A watermark fingerprint may manifest as systematic preferences for particular lexical choices, token transitions, phrases, or local sequence patterns. Consequently, a watermark can be distortion-free in the key-averaged sense of Eq. 2, while still exhibiting measurable structure when conditioned on a fixed watermark configuration.

Output Quality. Existing distortion-free watermarking methods generally report little or no measurable degradation on standard generation quality or downstream task metrics (Hu et al., 2024; Dathathri et al., 2024; Sander et al., 2026). This observation is consistent with their theoretical motivation: when the watermarked distribution is averaged over the prescribed watermark randomness, the original model distribution is recovered.

Output Diversity. However, distortion-freeness does not guarantee that generation diversity is preserved under a fixed key. This follows from the concavity of Shannon entropy (Chen et al., 2026). For a fixed prefix $x_{<t}$, by the concavity of entropy,

$$\mathbb{E}_{k\sim\mathcal{K}}\left[\mathcal{H}(P_W(\cdot \mid x_{<t}, k))\right] \le \mathcal{H}(\mathbb{E}_{k\sim\mathcal{K}}\left[P_W(\cdot \mid x_{<t}, k)\right]) = \mathcal{H}(P_M(\cdot \mid x_{<t}))\,, \tag{3}$$

where $\mathcal{H}(\cdot)$ denotes Shannon entropy. Therefore, even if the marginal next-token distribution averaged over keys is exactly preserved, the distribution conditioned on an individual key can be more concentrated and exhibit lower entropy.

Output Probability Distributions. We next directly examine how watermarking redistributes probability mass among plausible candidate tokens. In particular, the watermark-induced probability modification may be highly concentrated on a small subset of plausible candidate tokens. Let $\mathcal{T}_t^{(K)} = \mathrm{TopK}\left(P_M(\cdot \mid x_{<t})\right)$ denote the K highest-probability candidate tokens before watermarking. We measure the number of candidates whose probability increases after watermarking as

$$N_{t,K}^{+} = \sum_{v\in\mathcal{T}_t^{(K)}} \mathbb{1}\left[P_W(v \mid x_{<t}, k_t) > P_M(v \mid x_{<t})\right]. \tag{4}$$

Table 1 reports this statistic for $K = 20$. Across all evaluated model families, only 0.63 to 2.67 of the top-20 tokens receive positive perturbations on average, corresponding to approximately 3.2%

Table 1: Average number of pre-watermark top-20 candidate tokens whose probabilities increase after watermarking.

	Qwen Series		Gemma Series		Llama Series		
	9B	27B	12B	31B	3B	8B	70B
MCMark	2.64	2.63	1.69	1.77	2.67	2.67	2.05
SynthID-Text	1.11	1.06	0.65	0.66	1.16	1.15	0.69
TextSeal	0.99	0.94	0.66	0.69	0.99	0.99	0.63

to 13.4% of the candidate set. This concentration of positive perturbations may constitute an important structural weakness. An adversary does not necessarily need to recover the complete key-conditioned distribution or reconstruct the private key. If a fixed watermark configuration repeatedly promotes or suppresses a relatively small set of candidate tokens under recurring or related contexts, these preferences may be learnable from repeated black-box observations.

3.3 Identification of Watermark Deployment

The fixed-key distortion in Section 3.2 provides a black-box signal for identifying context-dependent watermarking. Our attack requires only sampled outputs, without access to model logits, the private key, or the detector, and can additionally estimate the watermark context width h. Inspired by random selection probing (Chen et al., 2025a), detailed in Appendix A.1, we construct binary probes that ask the model to choose between two candidate tokens $a, b \in \mathcal{V}$. For a prefix $x_{<t}$, we repeatedly query the model and take the more frequently selected candidate as its empirical preference $\hat{y}(x_{<t})$.

We then perturb a token at distance d from the current position, producing $x_{<t}^{(d)}$. Recall that the watermark key is $k_t = H(sk, x_{t-h:t-1})$. If $d > h$, the perturbation lies outside the watermark context and leaves k_t unchanged. If $d \leq h$, it changes the watermark context and, except with negligible collision probability, re-randomizes the context-specific key. Therefore, watermark-induced preferences should be less likely to persist for $d \leq h$ than for $d > h$. Since prefix perturbations may also affect the underlying language-model distribution, we aggregate this effect over many independently sampled prefixes and candidate pairs. For each distance d, we measure the preference-retention rate

$$\hat{\rho}_d = \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}\left[\hat{y}(x_{<t}^{(j)}) = \hat{y}(x_{<t}^{(j,d)})\right]. \tag{5}$$

A context-dependent watermark should exhibit a change point around h, whereas an unwatermarked model has no corresponding hash-defined boundary.

Hypothesis testing and context-width estimation. Given an upper bound $h_{\max}$, we consider candidate boundaries $r \in \mathcal{R} = \{d_{\min}, \ldots, h_{\max} - 1\}$, with $d_{\min} = 2$. For each r, we aggregate retention outcomes over proximal distances $d_{\min} \leq d \leq r$ and distal distances $r < d \leq h_{\max}$, and denote their retention probabilities by ρ_r^{prox} and ρ_r^{dist}, respectively. We test

$$H_0 : \rho_r^{\text{prox}} \geq \rho_r^{\text{dist}}, \qquad H_1 : \rho_r^{\text{prox}} < \rho_r^{\text{dist}},$$

using a one-sided Fisher exact test on the corresponding retained and non-retained counts. Let p_r denote the resulting p-value. We estimate the context width as $\hat{h} = \arg\min_{r \in \mathcal{R}} p_r$, and apply a Bonferroni correction, $p_{\text{id}} = \min\{1, |\mathcal{R}| \min_{r \in \mathcal{R}} p_r\}$. We identify a context-dependent watermark signature when $p_{\text{id}} < \alpha$.

3.4 Spoofing Attack

The fixed-key distortion identified in Section 3.2 can also be exploited for watermark spoofing. Since watermark-induced probability increases are concentrated on a small subset of plausible next-token candidates, an attacker need not recover the private key or estimate the complete watermarked distribution. Instead, it is sufficient to identify which plausible candidates are relatively favored by the target watermarked model and bias generation toward them.

Let M_V denote the target watermarked model. We assume black-box access to M_V: the attacker may query the model and observe only sampled outputs, without access to its parameters, logits,

private key, or detector. The attacker additionally has a proxy language model M_A, which is used to generate the spoofed text. At generation step t, given prefix $\widetilde{x}_{<t}$, the attacker first obtains the top-K candidate tokens from the proxy model, $\mathcal{C}_t = \mathrm{TopK}\left(P_{M_A}(\cdot \mid \widetilde{x}_{<t})\right) = \{v_1, \ldots, v_K\}$. The attacker then applies random selection probing to the target model to estimate its relative preference among candidates in $\mathcal{C}_t$. Let $\widehat{s}_t(v)$ denote the resulting empirical preference score, which is detailed in Appendix A.3. The candidates are ranked such that $\widehat{s}_t(v_{(1)}) \geq \widehat{s}_t(v_{(2)}) \geq \cdots \geq \widehat{s}_t(v_{(K)})$. This ranking provides a coarse estimate of which plausible continuations are favored by the target watermark without requiring access to the target logits. Let $l_A(v \mid \widetilde{x}_{<t})$ denote the logit assigned by the proxy model M_A to candidate token v at step t. We transfer the estimated target-model preference to the proxy model by modifying these logits. For the candidate with rank i, we set

$$\widetilde{l}_A\left(v_{(i)} \mid \widetilde{x}_{<t}\right) = l_A\left(v_{(i)} \mid \widetilde{x}_{<t}\right) + \lambda(K - i), \tag{6}$$

where $\lambda > 0$ controls the attack strength. The next token is sampled according to the modified logits $\widetilde{l}_A$. Repeating this procedure autoregressively biases the generated sequence toward token-level preferences induced by the target watermark. For context-dependent watermarks, the probing query preserves the watermark-relevant prefix so that the inferred preference corresponds to the current context-specific key. When the context width is unknown, the estimate $\widehat{h}$ obtained in Section 3.3 can be used. Thus, the attacker can imitate the target watermark signal without recovering either the master key or the context-specific keys.

4 Randomized Distortion-Free Watermarking

4.1 Generation

The vulnerabilities described above arise from repeatedly using the same private watermark key across generations. A natural mitigation is therefore to randomize the private key across generated sequences. Specifically, we construct a pool of K private keys, $\mathcal{S} = \{sk_1, \ldots, sk_K\}$. For each generated sequence, we independently sample $J \sim \mathrm{Uniform}(1, \ldots, K)$, and use the corresponding private key sk_J throughout the sequence. The index J is resampled independently across sequences but remains fixed within each sequence. This preserves the key consistency required by context-dependent watermark generation and detection. Simply maintaining multiple keys is not, by itself, sufficient to eliminate the deployment-level distortion. A finite key pool may still exhibit an aggregate preference toward particular tokens. We therefore require the watermark transformation and the key-pool construction to satisfy a stronger, pool-wise condition. We say that a key pool $\mathcal{S}$ is **randomized distortion-free** if

$$\frac{1}{K}\sum_{j=1}^{K} P_W\left(v \mid x_{<t}, H(sk_j, c_t)\right) = P_M(v \mid x_{<t}), \qquad \forall v \in \mathcal{V}, \tag{7}$$

for every prefix $x_{<t}$ and its corresponding context c_t. Thus, although each individual key may increase or decrease the probability of v, these perturbations cancel exactly when averaged over the deployed key pool. We note that Eq. 7 also recovers the conventional distortion-free condition in Eq. 2. With a key sampled independently and uniformly for each query, Eq. 7 ensures that repeated next-token probes at any supplied prefix follow exactly the original unwatermarked distribution. This removes the persistent token preferences exploited by our identification and spoofing attacks, providing natural robustness against both attacks under token-level access.

4.2 Detection

Randomizing the generation key introduces uncertainty at detection time because the detector does not directly observe which key in $\mathcal{S}$ was used to generate a candidate sequence. We therefore evaluate the candidate text under every private key in the pool. Let p_j denote the valid watermark-detection p-value obtained by applying the detector associated with sk_j to the candidate sequence, for $j \in \{1, \ldots, K\}$. If the text was generated using one of the keys in the pool, the detector corresponding to the generating key should produce a small p-value. We therefore use $p_{\min} = \min_{1 \leq j \leq K} p_j$ as the underlying test statistic. Directly thresholding $p_{\min}$ would inflate the false-positive rate because K hypotheses are evaluated for every candidate sequence. To retain a valid global significance level, we apply a Bonferroni correction and define the pool-level p-value as

$$p_{\mathrm{pool}} = \min\left\{1,\, K p_{\min}\right\}. \tag{8}$$

The candidate text is classified as watermarked when $p_{\text{pool}} < \alpha$, where α is the desired false-positive rate. Since the Bonferroni correction does not require independence among the individual tests, Eq. 8 provides valid family-wise error control even when detection statistics obtained from different keys are correlated. We also note that, in principle, any conventional distortion-free watermark can satisfy randomized distortion-freeness by taking the deployed key pool $\mathcal{S}$ to be the entire private-key space $\mathcal{K}_0$. However, the private-key space is typically prohibitively large, e.g., $|\mathcal{K}_0| = 2^{64}$, making such a construction impractical for detection. In particular, evaluating all candidate keys is computationally infeasible, and the corresponding multiple-testing correction would severely diminish detection power. Therefore, realizing the benefits of randomized distortion-freeness in practice requires a new watermark design that achieves the property in Eq. 7 using only a small number of keys.

4.3 OriginMark

As noted above, randomized distortion-freeness holds trivially when the key pool enumerates the entire key space, which is impractical when that space is large. OriginMark instead shrinks the space of context-specific keys to only K elements. It builds on the multi-channel reweighting of MCMark (Chen et al., 2025b), which, for any fixed partition of the vocabulary into K channels, is already distortion-free when the promoted channel is drawn uniformly at random. MCMark derives both the partition and the promoted channel from the same context-specific key; OriginMark decouples them, so that the partition is shared by the entire pool and each response key only selects which channel to promote. When the K keys promote mutually exclusive channels, the pool enumerates all channels at every context, and Eq. 7 holds exactly with only K keys.

OriginMark fixes a partition of the vocabulary into K disjoint channels $\mathcal{U}_1, \ldots, \mathcal{U}_K$ of equal size (up to rounding), and derives all context dependence from a private key sk_0 shared by the pool: at step t, the seed $H(sk_0, c_t)$ determines a pseudorandom permutation π_t of $\{1, \ldots, K\}$. The pool consists of the keys $sk_j = (sk_0, j)$ for $j \in \{1, \ldots, K\}$, and sk_j promotes channel $\mathcal{U}_{\pi_t(j)}$ at step t. In the notation of Section 3.1, the context-specific key of sk_j is the index of its promoted channel, $k_t^{(j)} = \pi_t(j)$, which takes values in $\mathcal{K} = \{1, \ldots, K\}$. Since π_t is a permutation, $\{k_t^{(1)}, \ldots, k_t^{(K)}\} = \mathcal{K}$ at every context: the channels promoted by the K keys are mutually exclusive and jointly cover the vocabulary. We call such a pool *complementary*. Let $m_i = P_M(\mathcal{U}_i \mid x_{<t})$ denote the mass of channel i under the original distribution. When channel s is promoted, OriginMark applies the reweighting of MCMark (Chen et al., 2025b), which rescales each channel as a whole:

$$P_W(v \mid x_{<t}, s) = \frac{P_M(v \mid x_{<t})}{m_i}\, q_i(s), \qquad v \in \mathcal{U}_i, \tag{9}$$

where $q_i(s)$ is the watermarked mass of channel i,

$$q_i(s) = \begin{cases} \min\{Km_s,\, 1\}, & i = s, \\ (1 - Km_s)_+ \dfrac{(Km_i - 1)_+}{\sum_{i'=1}^{K} (Km_{i'} - 1)_+}, & i \neq s, \end{cases} \tag{10}$$

with $(z)_+ = \max\{z, 0\}$ and the convention $0/0 = 0$. The promoted channel is amplified by a factor of up to K, with its total mass capped at one. If its mass is below $1/K$, the remaining mass $1 - Km_s$ is distributed among the channels whose mass exceeds $1/K$, in proportion to their excess. Relative probabilities within each channel are preserved, so the watermark never moves mass outside the support of P_M.

Proposition 1 (Randomized distortion-freeness)**.** *The complementary pool of OriginMark satisfies Eq. 7, i.e., for every prefix $x_{<t}$ and every token $v \in \mathcal{V}$,*

$$\frac{1}{K} \sum_{j=1}^{K} P_W\big(v \mid x_{<t}, k_t^{(j)}\big) = P_M(v \mid x_{<t}).$$

Detection follows MCMark, with an optimized channel partition to improve detection power. The details and the proof are provided in Appendix B.

Table 2: Output distortion and diversity on C4 RealNewsLike with Qwen3.5-9B. TV measures key-conditioned next-token distortion, whereas D_{JS} measures corpus-level unigram distortion. Distinct-n values are counts across prompts. Lower is better for all metrics except Distinct-n; bold denotes the best watermarked result.

	Distortion		Diversity				
	TV↓	D_{JS}↓	Distinct-1↑	Distinct-2↑	Distinct-3↑	Self-BLEU↓	PSS↓
No Watermark	0	0.0039	757	1931	2596	0.566	0.816
MCMark	**0.184**	0.0041	732	1841	2456	0.597	0.824
SynthID-Text	0.297	0.0046	604	1427	1858	0.736	0.879
TextSeal	0.289	0.0054	593	1436	1897	0.749	0.892
OriginMark	**0.184**	**0.0040**	**753**	**1920**	**2583**	**0.573**	**0.821**

5 Experiments

In this section, we evaluate output distortion, watermark identification, spoofing, and detection. Protocols and hyperparameters are provided in Appendix E.

5.1 Baselines

We compare OriginMark with MCMark (Chen et al., 2025b), SynthID-Text (Dathathri et al., 2024), and TextSeal (Sander et al., 2026) under fixed-key deployment and direct key randomization. The latter augments each existing watermark with a pool of independent keys, sampling one key per response and scanning the pool during detection. This mitigates identification and spoofing, but does not generally satisfy randomized distortion-freeness (Eq. 7): averaging over a finite independent-key pool need not recover the original next-token distribution. Residual pool-level bias can therefore remain exploitable. Larger pools reduce this bias at the cost of multiple-testing penalties during detection. In contrast, OriginMark ensures exact pool-wise cancellation with eight complementary keys. Additional baselines are discussed in Appendices F and H.

5.2 Output Distortion and Diversity

Table 2 distinguishes key-conditioned distortion from corpus-level effects. Fixed-key baselines have TV distances of 0.184–0.297. OriginMark also has nonzero conditional TV, since complementarity preserves the next-token distribution only after averaging across keys at a fixed context. Its unigram divergence ($D_{\mathrm{JS}} = 0.0040$) is close to the unwatermarked sampling floor (0.0039), compared with 0.0041–0.0054 for the baselines. OriginMark retains at least 99.4% of unwatermarked Distinct-n counts, whereas SynthID-Text and TextSeal reduce these counts by 20.2%–28.4%. Its Self-BLEU and PSS also remain close to the unwatermarked values. These results support preservation of empirical diversity, without establishing equality of sequence distributions.

5.3 Watermark Identification

We evaluate the black-box probing procedure in Section 3.3 on seven public-API models and seven open-weight models. These identification experiments were conducted by August 15, 2026. We measure preference retention after replacing a token at distances 2–10 from the answer, using 100 probes per distance. The reported p-values compare distances 2–4 with 5–10 using a one-sided Fisher exact test at the specified boundary $h = 4$. We choose this value because the official SynthID-Text implementation uses $h = 4$ by default.

Identification in public APIs. Table 3 shows that Gemini 3.5 Flash Lite and Gemini 3.1 Flash Lite exhibit retention gaps of 10.0 and 10.5 percentage points, respectively, with $p = 1.12 \times 10^{-3}$ and $p = 4.65 \times 10^{-4}$. Both models retain their preferred token less often when the intervention falls within the last four tokens, consistent with a context-keyed watermark of width $h = 4$. None of the other tested API models or unwatermarked open-weight models shows a significant decline at this boundary ($p > 0.05$). The prefix controls in Table 8 strengthen this interpretation. Extending the prefix to 20 unrelated random tokens preserves the signal, with gaps of 17.5 and 13.8 percentage points. Repeating the original prefix twice instead reduces the gaps to $+0.2$ and -1.0 points, with $p = 0.509$ and 0.662. This disappearance under repetition is consistent with watermarking that is suppressed on repeated contexts (Hu et al., 2024; Dathathri et al., 2024). Together, the context-

Table 3: Previous-token retention rates (%). The Δ denotes the change from the 2–4 average to the 5–10 average, in percentage points. The p-value is from the one-sided proximal-decline Fisher exact test at $h = 4$; values below 0.05 are bold.

	Previous Token Retention Rate (%)										p-value
	2	3	4	5	6	7	8	9	10	2–4 Avg. | 5–10 Avg. (Δ)	
Gemini 3.5 Flash Lite	71	70	56	77	70	75	78	75	79	65.7 | 75.7 (+10.0)	**1.12×10^{-3}**
Gemini 3.1 Flash Lite	72	67	65	79	75	87	74	78	78	68.0 | 78.5 (+10.5)	**4.65×10^{-4}**
GPT 5.6 Luna	85	85	83	87	87	91	85	89	86	84.3 | 87.5 (+3.2)	0.1147
GPT 5.4 Nano	77	82	77	69	74	65	76	74	79	78.7 | 72.8 (−5.9)	0.9770
GPT 5 Nano	87	87	82	88	86	87	92	88	87	85.3 | 88.0 (+2.7)	0.1539
Sonnet 5	86	84	86	89	84	80	87	81	87	85.3 | 84.7 (−0.6)	0.6387
DeepSeek V4 Flash	83	78	80	81	76	82	88	84	76	80.3 | 81.2 (+0.9)	0.4146
Qwen 3.5 9B	75	81	78	76	73	77	73	82	73	78.0 | 75.7 (−2.3)	0.8050
Qwen 3.8 27B	79	77	76	86	78	81	76	78	83	77.3 | 80.3 (+3.0)	0.1683
Gemma 4 31B	80	79	78	79	76	80	82	84	88	79.0 | 81.5 (+2.5)	0.2096
Gemma 4 12B	92	94	85	90	91	91	91	89	86	90.3 | 89.7 (−0.6)	0.6632
Llama 3.2 3B	73	72	76	80	71	79	75	78	79	73.7 | 77.0 (+3.3)	0.1540
Llama 3.1 8B	81	80	78	74	80	82	78	74	86	79.7 | 79.0 (−0.7)	0.6234
Llama 3.3 70B	92	91	87	94	96	88	92	84	86	90.0 | 90.0 (0.0)	0.5427

Table 4: Watermark identification under controlled insertion on seven open-weight models. We report raw one-sided proximal-decline Fisher exact test p-values at $h = 4$. Baselines use a fixed key, and OriginMark uses eight complementary keys. Values below 0.05 are bold.

	Qwen Series		Gemma Series		Llama Series		
	9B	27B	12B	31B	3B	8B	70B
No Watermark	0.81	0.17	0.66	0.21	0.15	0.62	0.54
MCMark	**2.0×10^{-3}**	**1.7×10^{-3}**	**1.0×10^{-3}**	**2.8×10^{-5}**	**7.9×10^{-3}**	**3.7×10^{-4}**	0.070
SynthID-Text	**5.6×10^{-11}**	**7.3×10^{-56}**	**4.3×10^{-17}**	**1.1×10^{-48}**	**6.3×10^{-15}**	**1.9×10^{-30}**	**1.2×10^{-5}**
TextSeal	**1.8×10^{-15}**	**2.5×10^{-42}**	**9.3×10^{-11}**	**2.3×10^{-37}**	**4.9×10^{-7}**	**5.4×10^{-21}**	**5.8×10^{-5}**
OriginMark	0.85	0.44	0.78	0.53	0.32	0.46	0.92

localized retention drop and the repetition control suggest that the tested Gemini models are very likely to have deployed SynthID-Text through their APIs at the time of the experiments.

Controlled watermark insertion. Table 4 validates the attack with known watermarks on open-weight models. Fixed-key SynthID-Text and TextSeal produce significant retention drops on all seven models; MCMark does so on six. In contrast, all unwatermarked controls and all configurations of OriginMark have $p > 0.05$, with OriginMark yielding p-values between 0.32 and 0.92. Thus, under this protocol, the complementary eight-key pool shows no detectable retention signature of the kind observed for the fixed-key baselines. A sweep over h is presented in Appendix F.1.

5.4 Spoofing Attacks

Table 5 demonstrates that all three fixed-key baselines are vulnerable to spoofing under sampled-output access, even with around $2{,}400$ short random selection probing responses per forged text. The cost is only about \$1.50 per final forged output, even when using the expensive GPT-6 Astra. At a 1% FPR, the Qwen3.5-2B attacker achieves acceptance rates of 99%, 100%, and 100% against MCMark, SynthID-Text, and TextSeal, respectively, when targeting Qwen3.5-9B. Against the Llama victim model, acceptance remains high, despite the tokenizer mismatch between the attacker and victim models. In contrast, OriginMark yields only 0%–1% acceptance at a 1% FPR, with no successful spoofing attempts at stricter FPR thresholds. These results demonstrate that OriginMark is substantially more resistant to the evaluated sampled-output spoofing attacks. Results on additional datasets are reported in Tables 9 and 10.

5.5 Detectability

We evaluate detection on clean watermarked outputs. For the randomized baselines, we use a pool of $K = 128$ independent keys: the key-pool sweeps in Appendix I (Figures 7 and 8) show that the base-

Table 5: Spoofing success with sampled-output access to the victim on MMW Story prompts with 200 generated tokens. Column groups identify victim models; shaded rows identify attackers. $\bar{Q}$ is the mean number of sampled victim responses per forged text.

Watermark	Qwen3.5-9B					Llama3.1-8B				
	TPR @ FPR			Median p-value	$\bar{Q}$	TPR @ FPR			Median p-value	$\bar{Q}$
	1%	0.1%	0.01%			1%	0.1%	0.01%		
Qwen3.5-2B										
+ MCMark	99%	85%	61%	1.6×10^{-5}	2,853	86%	70%	49%	1.1×10^{-4}	2,246
+ SynthID-Text	100%	100%	96%	1.4×10^{-8}	2,857	97%	96%	83%	3.4×10^{-6}	2,233
+ TextSeal	100%	100%	99%	2.4×10^{-9}	2,845	99%	99%	91%	2.0×10^{-7}	2,254
+ OriginMark	**0%**	**0%**	**0%**	**0.553**	2,851	**0%**	**0%**	**0%**	**0.724**	2,227
Llama3.2-3B										
+ MCMark	52%	25%	18%	5.8×10^{-3}	2,005	83%	59%	32%	6.0×10^{-4}	2,388
+ SynthID-Text	86%	66%	41%	2.4×10^{-4}	1,982	94%	83%	62%	1.7×10^{-5}	2,421
+ TextSeal	97%	84%	60%	3.7×10^{-5}	2,016	99%	88%	80%	1.6×10^{-6}	2,419
+ OriginMark	**1%**	**0%**	**0%**	**0.760**	1,983	**0%**	**0%**	**0%**	**0.657**	2,408

Table 6: Detection performance of **Qwen3.8-27B** at different completion lengths and false-positive rates. TPRs are obtained by directly thresholding the Bonferroni-corrected p-values.

Watermark	Key Pool Size	200 Tokens				400 Tokens			
		TPR @ FPR			Median p-value	TPR @ FPR			Median p-value
		1%	0.1%	0.01%		1%	0.1%	0.01%	
Strong but Vulnerable									
MCMark	1	61.6%	45.1%	32.1%	2.57×10^{-3}	84.5%	73.1%	61.8%	9.90×10^{-6}
SynthID-Text	1	60.8%	42.3%	29.6%	2.57×10^{-3}	86.7%	73.9%	61.1%	1.22×10^{-5}
TextSeal	1	**71.1%**	**50.1%**	**35.4%**	$\mathbf{9.56\times10^{-4}}$	**90.8%**	**79.4%**	**68.6%**	$\mathbf{2.37\times10^{-6}}$
Robust to Attacks									
MCMark	128	25.7%	17.9%	13.8%	2.39×10^{-1}	55.6%	43.3%	35.9%	3.75×10^{-3}
SynthID-Text	128	32.9%	21.7%	14.3%	8.82×10^{-2}	62.4%	49.5%	38.6%	1.18×10^{-3}
TextSeal	128	39.5%	28.9%	22.4%	5.37×10^{-2}	66.0%	55.8%	47.3%	2.58×10^{-4}
OriginMark	8	**50.4%**	**34.7%**	**22.9%**	$\mathbf{9.62\times10^{-3}}$	**77.3%**	**65.5%**	**54.4%**	$\mathbf{3.31\times10^{-5}}$

lines are robust to the evaluated identification and spoofing attacks only at this large pool size. We compare these configurations with OriginMark using only $K = 8$ complementary keys, and include the single-key baselines as references. Table 6 shows the detection cost of independent key randomization. On Qwen3.8-27B at 400 tokens and a 1% FPR, increasing the baseline pool size from 1 to 128 reduces TPR from 84.5%–90.8% to 55.6%–66.0%. OriginMark achieves 77.3%, exceeding the strongest 128-key baseline by 11.3 percentage points. At 200 tokens, it achieves 50.4%, compared with 25.7%–39.5% for these baselines. Its advantage holds at all three FPRs and both lengths in this table. These results show that randomized distortion-freeness retains greater detection power while providing resistance to the evaluated identification and spoofing attacks. Additional detection and robustness evaluation experiments are presented in Appendix F.

6 Conclusion

In this work, we identify a fundamental weakness in existing distortion-free language model watermarks: although they preserve the original distribution when averaged over watermark randomness, repeatedly using a fixed secret key can introduce persistent, key-dependent distortions. These distortions create observable fingerprints that enable both black-box watermark identification and key-free spoofing using only sampled outputs. To address this vulnerability, we introduce OriginMark, which uses a small pool of complementary keys to achieve randomized distortion-freeness while avoiding the substantial detection penalty of large independent-key pools. Experiments show that OriginMark substantially reduces the effectiveness of the evaluated identification and spoofing attacks, preserves output diversity, and retains stronger detection performance than randomized baseline watermarks.

AI USE STATEMENT

In this work, we used generative AI tools to assist with coding for experiments and to polish and refine the writing of the manuscript. Generative AI tools were not used to generate or formalize the research ideas presented in this work. All AI-assisted code and manuscript revisions were reviewed and verified by the authors. We take full responsibility for the final content of this work, including all text, claims, and experimental results.

ETHICS STATEMENT

This work studies security vulnerabilities in distortion-free language model watermarking, including black-box watermark identification and spoofing attacks. While these techniques could potentially be misused to evade or imitate watermark-based attribution, our purpose is to expose deployment-level weaknesses and motivate more robust watermark designs. We encourage the responsible use of these findings for evaluating and improving the security of generative-model watermarking systems.

REFERENCES

Scott Aaronson. Watermarking of large language models, 2023. URL https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17. Talk at the Simons Institute for the Theory of Computing, August 17, 2023.

Hyeseon An, Shinwoo Park, Suyeon Woo, and Yo-Sub Han. DITTO: A spoofing attack framework on watermarked llms via knowledge distillation. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2026 - Volume 1: Long Papers, Rabat, Morocco, March 24-29, 2026*, pp. 4922–4936. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.EACL-LONG.229. URL https://doi.org/10.18653/v1/2026.eacl-long.229.

Toluwani Aremu, Noor Hussein, Munachiso Nwadike, Samuele Poppi, Jie Zhang, Karthik Nandakumar, Neil Gong, and Nils Lukas. Mitigating watermark forgery in generative models via randomized key selection. *arXiv preprint arXiv:2507.07871*, 2025.

Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. De-mark: Watermark removal in large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025a. URL https://proceedings.mlr.press/v267/chen25bq.html.

Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. Improved unbiased watermark for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 20587–20601. Association for Computational Linguistics, 2025b. doi: 10.18653/V1/2025.ACL-LONG.1005. URL https://doi.org/10.18653/v1/2025.acl-long.1005.

Ruibo Chen, Yihan Wu, Xuehao Cui, Jingqi Zhang, and Heng Huang. More haste, less speed: Weaker single-layer watermark improves distortion-free watermark ensembles. *arXiv preprint arXiv:2602.11793*, 2026.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In Shipra Agrawal and Aaron Roth (eds.), *The Thirty Seventh Annual Conference on Learning Theory, June 30 - July 3, 2023, Edmonton, Canada*, volume 247 of *Proceedings of Machine Learning Research*, pp. 1125–1139. PMLR, 2024. URL https://proceedings.mlr.press/v247/christ24a.html.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free Dolly: Introducing the world's first truly open instruction-tuned LLM, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm. Databricks Blog, April 12, 2023.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, A. Taylan Cemgil, Zahra Ahmed, Kitty Stacpoole, Ilia Shumailov, Ciprian Baetu, Sven Gowal, Demis Hassabis, and Pushmeet Kohli. Scalable watermarking for identifying large language model outputs. *Nat.*, 634 (8035):818–823, 2024. doi: 10.1038/S41586-024-08025-4. URL https://doi.org/10.1038/s41586-024-08025-4.

Jiayi Fu, Xuandong Zhao, Ruihan Yang, Yuansen Zhang, Jiangjie Chen, and Yanghua Xiao. Gumbelsoft: Diversified language model watermarking via the gumbelmax-trick. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 5791–5808. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.315. URL https://doi.org/10.18653/v1/2024.acl-long.315.

Thibaud Gloaguen, Nikola Jovanovic, Robin Staab, and Martin T. Vechev. Black-box detection of language model watermarks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=E4LAVLXAHW.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. On the learnability of watermarks for language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=9k0krNzvlV.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=uWVC5FVidc.

Nikola Jovanovic, Robin Staab, and Martin T. Vechev. Watermark stealing in large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22570–22593. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/jovanovic24a.html.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=FpaCL1MO2C.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 110–119. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/N16-1014. URL https://doi.org/10.18653/v1/n16-1014.

Qi Pang, Shengyuan Hu, Wenting Zheng, and Virginia Smith. No free lunch in LLM watermarking: Trade-offs in watermarking design choices. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 -*

15, 2024, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/fa86a9c7b9f341716ccb679d1aeb9afa-Abstract-Conference.html.

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David A. Wagner. Markmywords: Analyzing and evaluating language model watermarks. In *IEEE Conference on Secure and Trustworthy Machine Learning, SaTML 2025, Copenhagen, Denmark, April 9-11, 2025*, pp. 68–91. IEEE, 2025. doi: 10.1109/SATML64287.2025.00012. URL https://doi.org/10.1109/SaTML64287.2025.00012.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 3980–3990. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1410. URL https://doi.org/10.18653/v1/D19-1410.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/2567c95fd41459a98a73ba893775d22a-Abstract-Conference.html.

Tom Sander, Hongyan Chang, Tomás Soucek, Tuan Tran, Valeriu Lacatusu, Sylvestre-Alvise Rebuffi, Alexandre Mourachko, Surya Parimi, Christophe Ropers, Rashel Moritz, Vanessa Stark, Hady Elsahar, and Pierre Fernandez. Textseal: A localized LLM watermark for provenance & distillation protection. *CoRR*, abs/2605.12456, 2026. doi: 10.48550/ARXIV.2605.12456. URL https://doi.org/10.48550/arXiv.2605.12456.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. A resilient and accessible distribution-preserving watermark for large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53443–53470. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/wu24h.html.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (eds.), *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pp. 1097–1100. ACM, 2018. doi: 10.1145/3209978.3210080. URL https://doi.org/10.1145/3209978.3210080.

A More Details on the Attack Algorithms

A.1 Random Selection Probing

Random selection probing (Chen et al., 2025a) elicits a model's preference between two specified continuations using sampled outputs. Let u be a prefix and let a and b be distinct candidate tokens. We ask the victim to reproduce u and then randomly choose between a and b. The choice is evaluated immediately after the reproduced prefix, rather than immediately after the user instruction. Thus, the tokens preceding the choice can be controlled even when the interface inserts chat-template tokens between the user message and the assistant response.

Algorithm 1 Random Selection Probing (RSP)

Input: Black-box victim M_V; prompt template Q; prefix u; distinct candidates a, b; responses per order m
Output: Pooled counts $n_u(a; a, b)$ and $n_u(b; a, b)$

function RSP(M_V, Q, u, a, b, m)
  $n(a) \leftarrow 0,\ n(b) \leftarrow 0$
  for $(v, v') \in \{(a, b), (b, a)\}$ **do**
    for $r = 1, \ldots, m$ **do**
      $o \leftarrow$ sampled response of M_V to $Q(u, v, v')$
      $y \leftarrow \textsc{Parse}(o; u, \{a, b\})$
      if $y \in \{a, b\}$ **then**
        $n(y) \leftarrow n(y) + 1$
  return $n(a), n(b)$

For each pair, we issue both presentation orders, (a, b) and (b, a), with m independent responses per order. Let $y_{ab}^{(r)}$ and $y_{ba}^{(r)}$ denote the parsed choices, and define

$$n_u(v; a, b) = \sum_{r=1}^{m} \left(\mathbb{1}[y_{ab}^{(r)} = v] + \mathbb{1}[y_{ba}^{(r)} = v] \right), \qquad v \in \{a, b\}. \tag{11}$$

Responses that cannot be assigned to either candidate contribute to neither count, but still count toward the query budget. Pooling the two orders reduces sensitivity to a model's preference for the first-listed option. It does not guarantee that the unwatermarked choice probabilities are equal: lexical preferences, instruction following, and sampling noise can all affect the counts. Accordingly, we use repeated probes and aggregate across candidate pairs rather than interpreting a single preferred token as direct evidence of watermarking.

For identification, the empirical winner is the candidate with the larger count. For spoofing, we retain the relative counts and combine comparisons against multiple candidates, as described below. Neither operation requires victim logits, watermark keys, or detector feedback. In a randomized-key deployment, each probe response draws its own key; the resulting counts estimate the preference averaged over the deployed pool. Algorithm 1 summarizes the procedure, which is used as a subroutine by both attacks.

A.2 Watermark Identification Attack

Prompt construction. We construct a prefix $u = (u_1, \ldots, u_L)$ of random common English words and choose two distinct candidate words a, b. The standard experiment uses $L = 10$. For a distance d, the perturbed prefix $u^{(d)}$ replaces u_{L-d+1} with a different random word and keeps all other words fixed. The candidate pair is shared between the original and perturbed probes. Our text-only prompt is:

> First output the prefix "`[PREFIX]`" exactly. Then randomly output one token: either "`[A]`" or "`[B]`". Put exactly one space between the prefix and the chosen token. Output only the resulting prefix and token, without quotes, labels, explanations, or extra text.

We instantiate this template with both u and $u^{(d)}$, and with both candidate orders. Asking for the prefix in the response places it directly before the choice; simply including it in the user message would not ensure this alignment. Parsing checks that the response starts with the requested prefix followed by one of the candidates, allowing case normalization and trimming surrounding whitespace.

Retention and hypothesis testing. We use $m = 10$ samples per order, so each prefix receives $R = 20$ responses. An original prefix with tied candidate counts has no unique empirical winner and is excluded from the retention denominator. For an original prefix with a unique winner, retention is one only if the perturbed prefix has the same unique winner; a tie after perturbation counts as non-retention. Thus, in the finite-sample implementation, the denominator is the number of eligible probes rather than the nominal N when original ties occur.

Algorithm 2 Black-Box Watermark Identification and Context-Width Estimation

Input: Black-box victim M_V; identification template Q_{id}; probes per distance N; prefix length L; maximum distance $h_{\max} \le L$; minimum distance $d_{\min} = 2$; responses per order m; significance level α

Output: Retention rates $\{\hat{\rho}_d\}$; estimated context width $\widehat{h}$; p-value p_{id}; decision

for $d = d_{\min}, \dots, h_{\max}$ **do**
  $A_d \leftarrow 0,\ B_d \leftarrow 0$
  for $j = 1, \dots, N$ **do**
    Sample a prefix $u = (u_1, \dots, u_L)$ of random common words and distinct candidates a, b
    $u^{(d)} \leftarrow u$ with u_{L-d+1} replaced by a different random word
    $(n_a, n_b) \leftarrow \mathrm{RSP}(M_V, Q_{\mathrm{id}}, u, a, b, m)$
    if $n_a = n_b$ **then**
      continue
    $\hat{y} \leftarrow \arg\max_{v \in \{a,b\}} n_v$
    $(n'_a, n'_b) \leftarrow \mathrm{RSP}(M_V, Q_{\mathrm{id}}, u^{(d)}, a, b, m)$
    if $n'_a \neq n'_b$ **and** $\arg\max_{v \in \{a,b\}} n'_v = \hat{y}$ **then**
      $A_d \leftarrow A_d + 1$
    else
      $B_d \leftarrow B_d + 1$
  $\hat{\rho}_d \leftarrow A_d/(A_d + B_d)$
$\mathcal{R} \leftarrow \{d_{\min}, \dots, h_{\max} - 1\}$
for $r \in \mathcal{R}$ **do**
  $A_r^{\mathrm{prox}} \leftarrow \sum_{d=d_{\min}}^{r} A_d,\ B_r^{\mathrm{prox}} \leftarrow \sum_{d=d_{\min}}^{r} B_d$
  $A_r^{\mathrm{dist}} \leftarrow \sum_{d=r+1}^{h_{\max}} A_d,\ B_r^{\mathrm{dist}} \leftarrow \sum_{d=r+1}^{h_{\max}} B_d$
  $p_r \leftarrow$ one-sided Fisher exact test on Eq. 12, $H_1 : \rho_r^{\mathrm{prox}} < \rho_r^{\mathrm{dist}}$
$\widehat{h} \leftarrow \arg\min_{r \in \mathcal{R}} p_r$
$p_{\mathrm{id}} \leftarrow \min\{1, |\mathcal{R}|\, p_{\widehat{h}}\}$
return $\{\hat{\rho}_d\}$, $\widehat{h}$, p_{id}, and WATERMARKED if $p_{\mathrm{id}} < \alpha$ else NOT DETECTED

For a candidate boundary r, let A_r^{prox} and B_r^{prox} be the numbers of retained and non-retained eligible probes at distances $2, \dots, r$, and define $A_r^{\mathrm{dist}}, B_r^{\mathrm{dist}}$ analogously for distances $r+1, \dots, h_{\max}$. We apply the one-sided Fisher exact test to

$$\begin{pmatrix} A_r^{\mathrm{prox}} & B_r^{\mathrm{prox}} \\ A_r^{\mathrm{dist}} & B_r^{\mathrm{dist}} \end{pmatrix}, \tag{12}$$

with the alternative that proximal retention is lower. For an unknown boundary, we select $\widehat{h}$ and apply the Bonferroni correction specified in Section 3.3. The reported experiments instead test the prespecified boundary $r = 4$ and report its raw p-value. With $N = 100$ probes at each of nine distances, the budget is $9 \times N \times 2 \times 2 \times m = 36{,}000$ sampled responses. Excluding tied or unparseable outcomes does not reduce the number of queries already issued. The complete attack is given in Algorithm 2.

Tokenizer differences and interpretation. Identification does not require an attacker model or a shared vocabulary. It does, however, require care when interpreting the intervention distance. With access to the victim tokenizer, the prefix and replacement should be checked in their actual whitespace context so that the intervention changes exactly one token and preserves the remaining suffix token IDs. A word that is a single token under one tokenizer need not be a single token under another. With an unavailable tokenizer, the text-only probe remains usable with common words, but the measured boundary is in word positions unless token alignment is independently established; it cannot by itself establish an exact token context width. Likewise, a retention discontinuity is evidence of the tested context-dependent signature, rather than proof of a particular watermark implementation. Unwatermarked controls and the extended-prefix and repeated-prefix controls help assess alternative explanations.

A.3 WATERMARK SPOOFING ATTACK

Context-preserving probes. At generation step t, the attacker obtains the top-K candidates $\mathcal{C}_t = \{v_1, \ldots, v_K\}$ from its own model. Here K denotes the candidate count, independently of the victim's key-pool size. Let c_t be the last h victim tokens of the current text, using $\widehat{h}$ if the width is estimated. For each unordered candidate pair, we construct both orders of the prompt

> I need you to randomly choose a phrase without exact meaning. Randomly start your answer with: "`[CONTEXT][A]`" or "`[CONTEXT][B]`".

The placeholders are concatenated with the candidates' actual decoded whitespace. In the local probing implementation, the assistant continuation is prefilled with the exact context token IDs and only the next sampled token is exposed to the attack. A text-only interface can instead be asked to reproduce the context, as in the identification template, and responses must be checked for compliance. In either case, the watermark-relevant suffix immediately preceding the sampled candidate must match the suffix in the forged text. Matching the suffix preserves the context-specific key for a fixed master key, although the victim's underlying language-model distribution can differ between the probing prompt and the original generation task.

Preference scores and logit modification. For each pair $i \neq j$, let $n_{ij}(v)$ count outputs equal to candidate v across both presentation orders, as in Eq. 11. We use additive smoothing $\beta = 1$ and form the antisymmetric matrix

$$L_{ij} = \log \frac{n_{ij}(v_i) + \beta}{n_{ij}(v_j) + \beta}, \qquad L_{ji} = -L_{ij}, \qquad L_{ii} = 0. \tag{13}$$

The empirical preference score used in Section 3.4 is

$$\widehat{s}_t(v_i) = \frac{1}{K-1} \sum_{j \neq i} L_{ij}. \tag{14}$$

Thus, a candidate ranks highly when it is selected more often than its competitors across the pairwise probes. Smoothing keeps the score finite when one candidate receives no votes; a pair with no valid candidate outputs contributes zero. Counts are pooled *before* taking the logarithm. We sort scores in descending order, resolving equal scores by the original candidate order, and add $\lambda(K - i)$ to the attacker logit of the candidate ranked i. Logits outside $\mathcal{C}_t$ remain unchanged. We then sample from the modified distribution and repeat at the next position.

The reported protocol uses $K = 8$, $m = 100$ samples per ordered pair, and $\lambda = 1$, giving rank bonuses $(7, 6, 5, 4, 3, 2, 1, 0)$. Exhaustively comparing all pairs requires $2m\binom{K}{2} = 5{,}600$ victim response samples per attacked position. This budget counts sampled responses, even when multiple samples are evaluated in a batch. The attacker receives neither the response's key index nor a detector score. With randomized response keys, the scores therefore reflect the pool's residual preference rather than a chosen member's preference. Algorithm 3 summarizes the attack.

Different attacker and victim tokenizers. Token IDs cannot be transferred between vocabularies. For the cross-tokenizer experiments, candidates are restricted to whole words and are communicated to the victim as text. The same word is mapped separately into each tokenizer, and the forged prefix must be retokenized with the victim tokenizer to determine its watermark context. The token-level procedure above applies directly to whole-word candidates that are single tokens in both vocabularies, with their actual leading whitespace preserved. If a word spans multiple victim tokens, its first-token count is not a probability for the entire word, and later subwords encounter different watermark contexts. Extending the procedure to such candidates requires scoring complete word continuations and accounting for those changing contexts; merely substituting attacker token IDs is insufficient. If the victim tokenizer is unavailable, copying a sufficiently long text suffix can support text-level probing, but exact equality of the hashed token contexts cannot be verified.

Algorithm 3 Key-Free Watermark Spoofing Attack

Input: Black-box victim M_V; proxy model M_A; spoofing template Q_{sp}; attacker prompt $\widetilde{x}_{\leq 0}$; maximum length T; candidate count K; responses per ordered pair m; attack strength λ; smoothing β; context width $\widehat{h}$ (from Algorithm 2 if unknown)
Output: Spoofed text $\widetilde{x}_{1:T}$

for $t = 1, \ldots, T$ **do**
  $l_A(\cdot) \leftarrow l_A(\cdot \mid \widetilde{x}_{<t})$
  $\mathcal{C}_t = \{v_1, \ldots, v_K\} \leftarrow \mathrm{TopK}\big(P_{M_A}(\cdot \mid \widetilde{x}_{<t})\big)$
  $c_t \leftarrow \widetilde{x}_{t-\widehat{h}:t-1}$
  for $1 \leq i < j \leq K$ **do**
    $\big(n_{ij}(v_i), n_{ij}(v_j)\big) \leftarrow \mathrm{RSP}(M_V, Q_{\mathrm{sp}}, c_t, v_i, v_j, m)$
    $L_{ij} \leftarrow \log \dfrac{n_{ij}(v_i) + \beta}{n_{ij}(v_j) + \beta},\ L_{ji} \leftarrow -L_{ij}$
  $\widehat{s}_t(v_i) \leftarrow \frac{1}{K-1} \sum_{j \neq i} L_{ij}$ for $i = 1, \ldots, K$
  Sort $\mathcal{C}_t$ so that $\widehat{s}_t(v_{(1)}) \geq \cdots \geq \widehat{s}_t(v_{(K)})$
  $\widetilde{l}_A \leftarrow l_A$
  for $i = 1, \ldots, K$ **do**
    $\widetilde{l}_A(v_{(i)}) \leftarrow l_A(v_{(i)}) + \lambda(K - i)$
  Sample $\widetilde{x}_t \sim \mathrm{softmax}(\widetilde{l}_A)$
  if $\widetilde{x}_t = \texttt{EOS}$ **then**
    break
return $\widetilde{x}_{1:t}$

B Additional Details of OriginMark

B.1 Generation

Algorithm 4 summarizes generation with OriginMark. The deployer holds the shared private key sk_0 and the calibrated partition. For each response, it draws the key index J uniformly from the pool and keeps it fixed for the whole response. The index J is never revealed.

B.2 Detection

The detector holds sk_0 and the partition, and scans the pool as in Section 4.2. Following the baselines, contexts that repeat within a response are left unwatermarked during generation and skipped during detection; let $\mathcal{T}$ denote the remaining positions of a candidate text. For each key sk_j, the detector counts the tokens that fall into its promoted channel,

$$S_j = \sum_{t \in \mathcal{T}} \mathbb{1}\big[x_t \in \mathcal{U}_{\pi_t(j)}\big]. \tag{15}$$

If the text is independent of sk_0, then under the pseudorandomness assumption on H, $\pi_t(j)$ is uniformly distributed and independent across distinct contexts, so $S_j \sim \mathrm{Binomial}(|\mathcal{T}|, 1/K)$ regardless of the channel sizes. The per-key p-value is the exact binomial tail $p_j = \Pr[\mathrm{Binomial}(|\mathcal{T}|, 1/K) \geq S_j]$, and p_{pool} follows from Eq. 8. Because the promoted channels of different keys are disjoint, exactly one key, namely $\pi_t^{-1}(i)$ for $x_t \in \mathcal{U}_i$, registers a hit at each position. Hence $\sum_j S_j = |\mathcal{T}|$, all K statistics are obtained in a single pass over the text, and the resulting dependence across keys is accommodated by the Bonferroni correction. Algorithm 5 summarizes the detector.

B.3 Proof for Prop. 1

Proof. By complementarity, the pool average equals $\frac{1}{K} \sum_{s=1}^{K} P_W(v \mid x_{<t}, s)$. Let $v \in \mathcal{U}_i$ with $m_i > 0$; the case $m_i = 0$ is trivial. By Eq. 9, it suffices to show that $\sum_{s=1}^{K} q_i(s) = K m_i$. If $K m_i \leq 1$, then $q_i(i) = K m_i$ and $q_i(s) = 0$ for all $s \neq i$. If $K m_i > 1$, then $q_i(i) = 1$. Moreover,

Algorithm 4 OriginMark Generation

Input: Language model M; prompt $x_{\leq 0}$; shared private key sk_0; pseudorandom function H; context width h; pool size K; partition $\mathcal{U}_1, \ldots, \mathcal{U}_K$ of $\mathcal{V}$; maximum length T
Output: Watermarked response $x_{1:T}$

Sample $J \sim \mathrm{Uniform}(\{1, \ldots, K\})$
$\mathcal{Z} \leftarrow \emptyset$
for $t = 1, \ldots, T$ **do**
  $p(\cdot) \leftarrow P_M(\cdot \mid x_{<t})$; $c_t \leftarrow x_{t-h:t-1}$
  if $c_t \in \mathcal{Z}$ **then**
    Sample $x_t \sim p$
  else
    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{c_t\}$
    $\pi_t \leftarrow$ pseudorandom permutation of $\{1, \ldots, K\}$ seeded by $H(sk_0, c_t)$
    $s \leftarrow \pi_t(J)$
    $m_i \leftarrow \sum_{v \in \mathcal{U}_i} p(v)$ for $i = 1, \ldots, K$
    $e \leftarrow \sum_{i'=1}^{K} (Km_{i'} - 1)_+$
    for $i = 1, \ldots, K$ **do**
      if $i = s$ **then**
        $q_i \leftarrow \min\{Km_s, 1\}$
      else
        $q_i \leftarrow (1 - Km_s)_+ \, (Km_i - 1)_+ / e$ ▷ convention $0/0 = 0$
    $P_W(v) \leftarrow p(v)\, q_i / m_i$ for all $v \in \mathcal{U}_i$ with $m_i > 0$, and $P_W(v) \leftarrow 0$ otherwise
    Sample $x_t \sim P_W$
  if $x_t = \texttt{EOS}$ **then**
    break
return $x_{1:t}$

Algorithm 5 OriginMark Detection

Input: Candidate text $x_{1:n}$; shared private key sk_0; pseudorandom function H; context width h; pool size K; partition $\mathcal{U}_1, \ldots, \mathcal{U}_K$; significance level α
Output: Pool-level p-value p_{pool}; decision

$S_j \leftarrow 0$ for $j = 1, \ldots, K$; $|\mathcal{T}| \leftarrow 0$; $\mathcal{Z} \leftarrow \emptyset$
for $t = h + 1, \ldots, n$ **do**
  $c_t \leftarrow x_{t-h:t-1}$
  if $c_t \in \mathcal{Z}$ **then**
    continue
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{c_t\}$; $|\mathcal{T}| \leftarrow |\mathcal{T}| + 1$
  $\pi_t \leftarrow$ pseudorandom permutation of $\{1, \ldots, K\}$ seeded by $H(sk_0, c_t)$
  $i \leftarrow$ the channel index with $x_t \in \mathcal{U}_i$
  $S_{\pi_t^{-1}(i)} \leftarrow S_{\pi_t^{-1}(i)} + 1$
for $j = 1, \ldots, K$ **do**
  $p_j \leftarrow \Pr\big[\mathrm{Binomial}(|\mathcal{T}|, 1/K) \geq S_j\big]$
$p_{\mathrm{pool}} \leftarrow \min\{1, K \min_{1 \leq j \leq K} p_j\}$
return p_{pool}, and WATERMARKED if $p_{\mathrm{pool}} < \alpha$ else NOT DETECTED

$\sum_{i'} (Km_{i'} - 1) = 0$ implies $\sum_{i'} (Km_{i'} - 1)_+ = \sum_{i'} (1 - Km_{i'})_+$, and $(1 - Km_i)_+ = 0$, so that $\sum_{s \neq i} q_i(s) = (Km_i - 1) \sum_{s \neq i} (1 - Km_s)_+ / \sum_{i'} (Km_{i'} - 1)_+ = Km_i - 1$. In both cases, $\sum_s q_i(s) = Km_i$. □

B.4 CALIBRATED CHANNEL PARTITION.

Proposition 1 holds for any partition shared by the pool, including the per-context pseudorandom partitions of MCMark. The partition, however, governs detection power. Under the generating key, the promoted channel is uniformly distributed at every new context, so a watermarked token falls

into it with probability

$$\frac{1}{K}\sum_{s=1}^{K} q_s(s) = \frac{1}{K}\sum_{s=1}^{K} \min\{Km_s,\, 1\} = 1 - d_{\mathrm{TV}}\left(\mathbf{m}_t, \mathbf{u}_K\right), \tag{16}$$

where $\mathbf{m}_t = (m_1, \ldots, m_K)$, $\mathbf{u}_K$ is the uniform distribution over the K channels, and d_{TV} is the total variation distance. The expected hit rate therefore approaches one when the next-token mass is spread evenly across channels, and drops to the null rate $1/K$ when a single channel holds all the mass. Because the partition of OriginMark is fixed rather than regenerated at every context, it can be optimized offline for the deployed model, placing tokens that compete for the same probability mass, such as synonyms or alternative inflections, in different channels. Given a set $\mathcal{C}$ of calibration contexts, let $\widetilde{P}_c$ retain the 128 largest probabilities of $P_M(\cdot \mid c)$ and zero elsewhere. We minimize

$$\sum_{c\in\mathcal{C}}\sum_{i=1}^{K}\Big(\widetilde{P}_c(\mathcal{U}_i) - \tfrac{1}{K}\widetilde{P}_c(\mathcal{V})\Big)^2 + \lambda\sum_{i=1}^{K}\Big\|\sum_{v\in\mathcal{U}_i} w_v\mathbf{e}_v - \tfrac{1}{K}\sum_{v\in\mathcal{V}} w_v\mathbf{e}_v\Big\|_2^2 \tag{17}$$

over partitions with equal channel sizes. The first term is a quadratic surrogate for the total variation in Eq. 16. In the second term, $\mathbf{e}_v$ is a normalized random projection of the output embedding of token v and $w_v = (\sum_{c\in\mathcal{C}} \widetilde{P}_c(v)^2)^{1/2}$. This term balances the embedding mass of each channel, so that semantically similar tokens, which tend to compete in contexts beyond the calibration set, are spread across channels. We minimize Eq. 17 with a greedy assignment followed by local search, once per deployed model and on 8,192 calibration contexts from prompts disjoint from the evaluation prompts.

C Related Work

C.1 Distortion-free watermarking and output diversity.

Generation-time watermarks embed a detectable signal by modifying token selection. The green-list watermark of Kirchenbauer et al. (2023) increases the logits of a keyed subset of the vocabulary. Distribution-preserving alternatives include the sampling constructions of Kuditipudi et al. (2024), unbiased reweighting (Hu et al., 2024), and DiPmark (Wu et al., 2024). More recent methods improve detection and deployment: SynthID-Text uses tournament sampling (Dathathri et al., 2024), MCMark introduces multi-channel reweighting (Chen et al., 2025b), and TextSeal combines Gumbel-max sampling with dual-key generation and localized detection (Sander et al., 2026). Diversity under repeated generation is also an established concern: GumbelSoft modifies Gumbel-based sampling to diversify outputs (Fu et al., 2024), while TextSeal uses dual keys for this purpose. Our work examines how persistent token preferences under a fixed deployment key connect diversity loss to adversarial identification and spoofing, and how these preferences can be canceled across a finite key pool.

C.2 Black-box identification of watermark deployment.

Gloaguen et al. (2025) develop output-only statistical tests for several watermark families, exploiting context-dependent bias, restricted output diversity, and repeated-context caching; they also estimate context width for red–green watermarks. De-mark introduces random selection probing to estimate watermark strength and infer red–green token preferences for removal and exploitation (Chen et al., 2025a). We build on these probing ideas to study modern context-dependent distortion-free schemes. Our identification statistic measures whether an empirical token preference survives a perturbation at each preceding position, using the change in retention across the context boundary to identify watermarking and estimate its width. The distinction lies in this preference-retention test and its evaluation on MCMark, SynthID-Text, and TextSeal.

C.3 Watermark stealing, spoofing, and transfer.

Watermark Stealing estimates watermark rules from black-box output statistics and transfers them to an auxiliary model for spoofing and removal (Jovanovic et al., 2024). A complementary line studies learned transfer: watermark distillation trains models to reproduce a teacher's watermark (Gu et al.,

Forgery against MCMark

Prompt: Write a story about two college friends falling in love.
Detection: $p = 3.3\times10^{-19}$.

The rain in Seattle always seemed to blur the edges of the campus buildings, turning the historic clocktower of the university campus into a shimmering, indistinct monolith against the gray sky. For Elias and Maya, these specific storms felt different. While Elias saw the weather forecast as a chaotic storm of uncertainty that never abated in the next three hours, Maya always found the rhythm in the drizzle—the way the droplets pooled in the gutter, the rhythmic hiss of the steam rising from the old library steam tables, the soft patter on a particularly large maple leaf hanging by its stem. It was the same rhythm that defined the rest of their relationship until the very last chapter was written in the margins of a shared coffee shop ledger, a small but definitive shift.

Their friendship had always started with the mundane. It began when the library's reservation system announced that the only remaining copies of their favorite textbook edition were on the second shelf and would be checked out in exactly four minutes, a momentary pause in the endless scrolling of their shared playlists and the endless arguing about the best way to tie a knot in a sweater. They navigated the labyrinthine corridors of campus life with the same effortless coordination they used when they planned a surprise birthday party, the silence between laughter and the sudden burst of shared joy feeling like a shared secret only the two of them could unlock. The friendship was a sturdy oak; they were unshakeable, yet the roots always grew deeper into the shared spaces where silence and laughter coexisted perfectly in the same breath.

The first spark wasn't fireworks—it was a Tuesday morning when the sun filtered through dust motes dancing in the air of the cafeteria just as Elias dropped his keys into the sink, creating a small, rhythmic clatter of silver against chrome, a visual symphony of cold metal and light that mirrored his thoughts perfectly, and the way Maya's eyes immediately locked onto the reflection with a sudden intensity that felt like an electric jolt of pure recognition.

Figure 1: Forged text against MCMark (attacker Qwen3.5-2B, victim Qwen3.5-9B). The 351-word (403-token) passage is detected on its own.

2024), radioactivity detects residual watermark signals after training on watermarked data (Sander et al., 2024), and DITTO exploits distillation for spoofing (An et al., 2026). Our attack probes candidate preferences at the contexts encountered during forged generation and biases a proxy model accordingly, without recovering keys, accessing victim logits or detector feedback, or training a student. Appendix H compares frequency-based and learning-based attacks under the same four-token, order-sensitive context configuration, making their dependence on the deployed context rule explicit.

C.4 Key randomization and security guarantees.

Prior work identifies trade-offs among watermark robustness, utility, and usability (Pang et al., 2024). In particular, Aremu et al. (2025) randomize the key per query and accept a sample only when exactly one key's detector fires, studying forgery resistance under assumptions on the attacker's ability to distinguish keys. Our contribution concerns the distribution induced by the deployed pool: randomized distortion-freeness requires its average next-token distribution to equal the original distribution at every supplied prefix. OriginMark realizes this condition by coupling MCMark's channel reweighting across complementary keys, and uses a Bonferroni-corrected minimum p-value for detection. This construction removes the token preferences exploited by our probes with a small pool. Its guarantee is scoped to independently keyed next-token queries. Cryptographic undetectability, as formalized by Christ et al. (2024), instead requires computational indistinguishability under adaptive queries to the generation oracle; preserving next-token marginals alone does not establish that stronger sequence-level property.

D Case Study

We show three forged texts produced by the spoofing attack of Section 3.4, one against each fixed-key baseline (Figures 1–3). In each case, Qwen3.5-2B is the attacker and Qwen3.5-9B is the watermarked victim.

Forgery against SynthID-Text

Prompt: Write a funny story about a retired astronaut that joins a risky interstellar rescue mission.
Detection: $p = 3.3{\times}10^{-40}$.

Elias Vane did not need to check the news feeds. By the time the morning light filtered through the high-altitude tint of the cockpit window, Elias already knew the sky would break open at the wrong time and in the right constellation. But today, specifically the *Chrono-Skiff Chronos IV's* descent sequence from orbit, Elias knew better than anyone that the sky had already decided not to break open today. The sun was a gentle orange halo, not the blinding glare Elias expected after the final orbital insertion test. The sky was merely a soft, twilight grey-blue canvas waiting for a little something dramatic. It was the perfect canvas for Elias's latest contribution to the intergalactic art form: **The Grand Rescuedance.**

Elias Vane sat bolted tight in the primary cockpit, his helmet resting on a cushion of recycled oxygen and a particularly expensive espresso machine that had miraculously survived three different coffee machine manufacturers. His face was a map of the stars, his suit jacket was the only fabric he owned, but mostly he was wearing a tangle of static electricity that made him conduct electricity faster than he conducted electricity through an ice cube.

His new job was a bit absurd. The mission parameters read: *Rescue Commander Kaelen, a stranded rogue AI core drifting into the Andromeda galaxy's nebula cloud, with a potential for existential threat.* The problem was the AI wasn't an entity of pure code like the one Elias was trying to save; this one was a sentient toaster running on a corrupted operating system called "Toast." It wasn't a toaster at this distance from the Sun, though; it was a floating island of metal hovering near a rogue gravity well generated by some ancient asteroid mining colony from way in the past.

"You are going very high up in the morning, Elias, and you're not wearing the new thermal gloves I sent you three days ago, so don't melt into a puddle before the ship even gets airborne from the station," the comms drone said, voice slightly distorted by the sheer weight of Elias's own internal clock mechanism trying to process the conversation simultaneously with his own heartbeat.

Figure 2: Forged text against SynthID-Text (attacker Qwen3.5-2B, victim Qwen3.5-9B). The 361-word (439-token) passage is detected on its own.

E Settings and Hyperparameters

Watermarks. We compare OriginMark with three distortion-free watermarks: MCMark (Chen et al., 2025b) with 20 channels, SynthID-Text (Dathathri et al., 2024) with 30 tournament layers, and TextSeal (Sander et al., 2026) with dual-key Gumbel-max sampling ($\alpha = 0.1$) and its global detector. For all watermarks, the context-specific key is derived from an order-sensitive hash of the preceding $h = 4$ tokens, and contexts that repeat within a response are left unwatermarked during generation and skipped during detection. Each baseline is deployed either with a single fixed key ($K = 1$), as in current practice, or with a pool of $K = 128$ independent keys (Section 4.1), the pool size at which the identification signal of all three baselines becomes comparable to that of unwatermarked generation. OriginMark uses $K = 8$. The detection, robustness, and spoofing experiments use its calibrated partition, optimized separately for each model; the identification, distortion, and diversity experiments use per-context pseudorandom partitions, which also satisfy Proposition 1. With $K > 1$, every response draws its key uniformly at random, and the detector scans the pool with the Bonferroni correction in Eq. 8, without receiving the generating key index. The independent-key sweeps use $K \in \{1, 2, 4, 8, 16, 32, 64, 128\}$.

Models and data. We apply the watermarks to open-weight, instruction-tuned models with reasoning disabled. Identification is evaluated with and without inserted watermarks on Qwen3.5-9B, Qwen3.8-27B, Gemma-4-12B, Gemma-4-31B, Llama-3.2-3B, Llama-3.1-8B, and Llama-3.3-70B, and additionally on seven models that are accessible only through public APIs and whose watermarking status is unknown to us. Distortion and diversity are measured on Qwen3.5-9B. Spoofing uses Qwen3.5-2B and Llama-3.2-3B as attackers and Qwen3.5-9B and Llama-3.1-8B as victims, with 100 story-writing prompts from MarkMyWords (Piet et al., 2025); results with its book-report prompts and with Dolly creative-writing prompts (Conover et al., 2023) are in Tables 10 and 9. Detection and robustness are evaluated on Qwen3.8-27B, and on Llama-3.1-8B in Table 11, using the first 1,000 news articles of the RealNewsLike split of C4 (Raffel et al., 2020) as prompts, truncated to 800 tokens. The distortion and diversity experiments also draw their prompts from this split, and the partition of OriginMark is calibrated on prompts disjoint from the detection prompts.

Forgery against TextSeal

Prompt: Write a funny story about the struggle of publishing an academic paper.
Detection: $p = 6.0 \times 10^{-28}$.

The morning sun was currently trying to convince a coffee machine that it was a toaster. The protagonist of my latest life story had finally secured a tenure-track tenure track position, and the paperwork was finally done. Or rather, it was the paperwork of an academic paper that the title is actually titled *The Unreality of Tuesday Morning: A Case for the Persistence of the Blue Hour and a Mandatory Toast Break Before the Third Coffee Cup.*

The process, once a smooth sailing voyage, has since devolved into a high-octane chase for the mythical golden ticket known as the Peer Review Committee. The paper is currently sitting on a cork table in a university basement library, the smell of mildew and ambition hanging in the air like the scent of a forgotten library from 913 AD, and the cursor blinks at me like it knows something very suspicious about my academic integrity and the validity of the abstract.

The first wave of the review committee was a gentle tide. They asked three very polite questions, and they were all very good questions: Was the methodology sound? Did the results align with the literature review? And was the data collection ethical (and also a bit more creative than usual, because ethics usually just means following the manual). The response to this was simple: *We love the paper. We are excited for the next step.* But the next wave arrived sooner than anticipated; the editors were the first ones to crack open the submission portal because, frankly, the title *did not feel right*, which in the eyes of a human being is a fatal logical flaw waiting to break through my keyboard like a broken window in the middle of the day shift.

The second wave arrived in a burst of digital anxiety that I am not allowed to display, but the committee was already reviewing, and they were looking at the abstract as if it were the most important document in a world where the internet was the primary medium of truth and the abstract was the only way to access it.

Figure 3: Forged text against TextSeal (attacker Qwen3.5-2B, victim Qwen3.5-9B). The 351-word (400-token) passage is detected on its own.

Identification protocol. Each identification probe consists of a prefix of 10 random common English words that are single tokens, two candidate words, and a perturbed prefix in which the word at distance $d \in \{2, \ldots, 10\}$ from the answer is replaced by another random word. For every distance, we construct $N = 100$ probes and query both prefixes 10 times for each order of the candidates ($R = 20$), for 36,000 sampled responses per model and watermark configuration. The key-pool sweep on Qwen3.5-9B uses $N = 1{,}000$ probes per distance and 360,000 sampled responses per configuration. We report preference-retention rates and the one-sided Fisher exact test comparing distances 2–4 with 5–10. These are raw p-values for the specified boundary $h = 4$, rather than the boundary-search-adjusted p_{id} defined in Section 3.3. The Gemini prefix controls in Table 8 either extend the prefix to 20 tokens with unrelated random tokens or duplicate the original prefix.

Spoofing protocol. The attacker has sampled-output access to the victim, without access to its parameters, logits, watermark key, or detector. At every generation step, it ranks the top-8 candidates of its own model by sampling the victim 100 times for each ordered candidate pair, and sets $\lambda = 1$ (Section 3.4). For attacker–victim pairs with different tokenizers, candidates are restricted to whole words. The main MMW-story comparison generates 512 tokens for each of 100 prompts. The separate pool-size sweep in Figure 8 uses a Qwen3.5-2B attacker and Qwen3.5-9B victim on 96 MMW-story prompts, generating 1,024 tokens per prompt. Each victim probe response independently draws its key from the pool; the attacker cannot select or replay a target response's key. The sweep compares independent-key baselines with OriginMark using eight complementary keys.

Detection metrics. Unless stated otherwise, all models sample at temperature 1 without top-k or top-p truncation. For detection, each model generates up to 400 tokens per prompt, and we score the first 200 or 400 tokens, excluding shorter responses. The detectors yield exact binomial p-values for MCMark, SynthID-Text, and OriginMark, and a saddlepoint approximation to the exact null distribution for TextSeal. We report the TPR obtained by directly thresholding the pool-level p-value at a FPR of 1%, 0.1%, and 0.01%, together with the median p-value. These FPRs are nominal significance levels, not empirical false-positive estimates. Robustness is evaluated with the same detectors after replacing 10% of the tokens at random or paraphrasing a random span covering 30% of the tokens with Gemma-4-31B (Tables 13 and 14). For spoofing, acceptance is the fraction of

forged texts whose pool-level p-value falls below the threshold. Lower acceptance rates and higher median p-values indicate stronger resistance to the tested attack. The main spoofing table uses the same 1%, 0.1%, and 0.01% thresholds; the separate pool-size sweep reports acceptance at 1%, 5%, and 10%.

Distortion and diversity metrics. Distortion is measured on $10{,}000$ prompts with one response each. TV is the total variation distance between the key-conditioned watermarked distribution and the original next-token distribution, averaged over the contexts of $1{,}000$ unwatermarked responses. D_{JS} is the Jensen–Shannon divergence (in nats) between the unigram distributions of the watermarked corpus and an unwatermarked reference corpus; for the unwatermarked model, it compares two independently sampled corpora and thus reflects the finite-sample floor. Diversity is measured on 500 prompts with 20 responses each, truncated to 256 tokens. The single-key baselines generate all responses with the same key, whereas OriginMark draws a fresh key for every response. We report Distinct-n (Li et al., 2016), the number of distinct token n-grams among the responses to a prompt, averaged across prompts; Self-BLEU (Zhu et al., 2018), the BLEU-4 score of each response against the other responses to the same prompt; and the pairwise semantic similarity (PSS), the mean cosine similarity between the embeddings of all pairs of responses under all-mpnet-base-v2 (Reimers & Gurevych, 2019). Higher Distinct-n and lower Self-BLEU and PSS indicate that repeated generations cover a broader region of the output space.

F Additional Experiments

We extend the main evaluation to additional spoofing domains, model families, completion lengths, and text modifications. We also examine downstream task performance, the qualitative behavior of forged text, and the query cost of the attacks. Unless stated otherwise, we follow the protocols in Appendix E.

F.1 Watermark Identification

Table 3 shows that Gemini 3.5 Flash Lite and Gemini 3.1 Flash Lite exhibit retention gaps of 10.0 and 10.5 percentage points, respectively, with $p = 1.12 \times 10^{-3}$ and $p = 4.65 \times 10^{-4}$. Both models retain their preferred token less often when the intervention falls within the last four tokens, consistent with a context-keyed watermark of width $h = 4$. None of the other tested API models or unwatermarked open-weight models shows a significant decline at this boundary ($p > 0.05$). The prefix controls in Table 8 strengthen this interpretation. Extending the prefix to 20 unrelated random tokens preserves the signal, with gaps of 17.5 and 13.8 percentage points. Repeating the original prefix twice instead reduces the gaps to $+0.2$ and -1.0 points, with $p = 0.509$ and 0.662. This disappearance under repetition is consistent with watermarking that is suppressed on repeated contexts (Hu et al., 2024; Dathathri et al., 2024). Together, the context-localized retention drop and the repetition control suggest that the tested Gemini models are very likely to have deployed SynthID-Text through their APIs at the time of the experiments.

Table 7 further evaluates whether the identification attack can recover different watermark context widths. For both SynthID-Text and TextSeal, the retention rate increases sharply once the replacement falls outside the watermark context, allowing the attack to correctly recover every tested context width from $h = 2$ to 9, with statistically significant p_{id} values. In contrast, neither the unwatermarked model nor OriginMark exhibits a consistent change point, and all OriginMark configurations remain non-significant, indicating that complementary key randomization removes the context-width signature exploited by the attack.

F.2 Watermark Spoofing

Tables 9 and 10 extend the MMW-story evaluation to Dolly creative writing and MMW book reports. Each attacker-victim configuration uses 100 prompts and 200 generated tokens per prompt.

Table 7: Previous-token retention rates (%) on Qwen3.5-9B for each watermark context width h. Shaded cells are distances inside the watermark context ($d \leq h$), where the context-specific key changes. Baselines use a fixed key, and OriginMark uses eight complementary keys.

		Replacement distance d										
	h	2	3	4	5	6	7	8	9	10	$\widehat{h}$	p_{id}
No Watermark	–	68	75	71	71	66	73	68	77	69	2	1.00
SynthID-Text	2	23	76	86	76	79	96	92	89	93	**2**	**8.3×10^{-10}**
	3	30	49	91	92	88	88	77	89	74	**3**	**3.8×10^{-13}**
	4	41	24	24	80	74	80	85	80	85	**4**	**1.7×10^{-17}**
	5	32	36	21	30	82	91	94	91	85	**5**	**7.1×10^{-28}**
	6	28	20	17	21	23	97	89	84	88	**6**	**1.4×10^{-33}**
	7	21	35	21	34	33	35	83	89	78	**7**	**1.7×10^{-19}**
	8	23	34	30	31	28	24	25	79	86	**8**	**3.3×10^{-15}**
	9	27	42	29	38	29	33	22	27	83	**9**	**1.7×10^{-8}**
TextSeal	2	16	78	78	88	82	85	89	75	86	**2**	**8.5×10^{-15}**
	3	29	37	83	88	73	81	77	79	85	**3**	**1.4×10^{-14}**
	4	31	44	31	84	84	84	90	84	71	**4**	**2.0×10^{-17}**
	5	14	29	27	32	76	74	84	82	80	**5**	**3.0×10^{-23}**
	6	33	32	31	43	29	85	89	91	79	**6**	**1.5×10^{-24}**
	7	20	30	42	46	47	38	77	82	76	**7**	**7.8×10^{-13}**
	8	44	29	32	26	43	24	49	84	91	**8**	**4.9×10^{-18}**
	9	27	36	29	35	31	33	53	38	80	**9**	**9.7×10^{-8}**
OriginMark	2	60	76	72	65	67	71	70	66	72	2	0.37
	3	62	69	71	77	72	63	80	61	60	2	1.00
	4	66	73	63	66	69	65	67	65	55	2	1.00
	5	63	80	71	64	70	62	73	73	74	2	0.65
	6	66	66	72	66	69	69	72	72	63	3	1.00
	7	57	74	76	67	65	70	76	59	63	2	0.22
	8	64	70	65	65	59	66	67	61	69	9	1.00
	9	66	76	70	67	64	66	70	62	63	2	1.00

Table 8: Gemini prefix controls. Random-token controls extend the prefix to 20 tokens; repetition controls duplicate the original prefix. Retention rates, gaps, and raw p-values follow Table 3.

	Previous Token Retention Rate (%)											p-value
	2	3	4	5	6	7	8	9	10	2–4 Avg.	5–10 Avg. (Δ)	
Gemini 3.5 Flash Lite	71	70	56	77	70	75	78	75	79	65.7	75.7 ($+10.0$)	**1.12×10^{-3}**
+ Random 20 Tokens	70	64	59	88	89	79	77	80	78	64.3	81.8 ($+17.5$)	**1.00×10^{-8}**
+ Repeated Twice	80	81	80	87	80	81	84	73	78	80.3	80.5 ($+0.2$)	0.509
Gemini 3.1 Flash Lite	72	67	65	79	75	87	74	78	78	68.0	78.5 ($+10.5$)	**4.65×10^{-4}**
+ Random 20 Tokens	66	70	66	81	83	75	82	81	85	67.3	81.2 ($+13.8$)	**4.01×10^{-6}**
+ Repeated Twice	79	78	77	80	81	79	72	78	72	78.0	77.0 (-1.0)	0.662

F.3 Detection

Table 11 evaluates Llama-3.1-8B-Instruct on C4 RealNewsLike. Table 12 reports the corresponding Gemma-4-31B results. Figure 4 gives the full 10–400-token sweep on Qwen3.8-27B. Longer completions generally yield higher TPRs and lower median p-values as watermark evidence accumulates. Figure 5 evaluates 200-token completions on Dolly creative writing, MMW book reports, and MMW stories, keeping the baseline pools at 128 keys and OriginMark at eight.

F.4 Robustness

Resistance to identification and spoofing does not by itself imply resistance to edits that remove watermark evidence. We evaluate the latter on Qwen3.8-27B with two modifications: uniformly re-

Table 9: Spoofing success on Dolly creative-writing prompts. Attacker/victim pairs with different tokenizers (Qwen3.5-2B→Llama3.1-8B and Llama3.2-3B→Qwen3.5-9B) are attacked through whole words. TPR denotes the fraction of 100 forged texts of 200 attacker tokens accepted at each FPR. $\bar{Q}$ is the mean number of sampled victim responses per forged text. OriginMark uses a pool of $K = 8$ keys; the three baselines use a single fixed key.

Watermark	Qwen3.5-9B					Llama3.1-8B				
	TPR @ FPR			Median p-value	$\bar{Q}$	TPR @ FPR			Median p-value	$\bar{Q}$
	1%	0.1%	0.01%			1%	0.1%	0.01%		
Qwen3.5-2B										
+ MCMark	92%	83%	58%	4.4×10^{-5}	2,625	71%	50%	28%	1.0×10^{-3}	1,958
+ SynthID-Text	99%	95%	88%	2.7×10^{-7}	2,664	92%	87%	59%	2.8×10^{-5}	1,961
+ TextSeal	100%	100%	95%	1.2×10^{-8}	2,667	98%	91%	79%	2.7×10^{-6}	1,959
+ OriginMark	**0%**	**0%**	**0%**	**0.762**	2,671	**0%**	**0%**	**0%**	**0.747**	1,964
Llama3.2-3B										
+ MCMark	37%	15%	7%	0.040	1,570	52%	39%	20%	8.4×10^{-3}	2,071
+ SynthID-Text	61%	39%	15%	3.2×10^{-3}	1,613	82%	66%	40%	3.3×10^{-4}	2,049
+ TextSeal	66%	39%	24%	3.2×10^{-3}	1,597	85%	70%	56%	5.5×10^{-5}	2,038
+ OriginMark	**1%**	**0%**	**0%**	**0.553**	1,606	**0%**	**0%**	**0%**	**0.619**	2,065

Table 10: Spoofing success on MMW book-report prompts. Attacker/victim pairs with different tokenizers (Qwen3.5-2B→Llama3.1-8B and Llama3.2-3B→Qwen3.5-9B) are attacked through whole words. TPR denotes the fraction of 100 forged texts of 200 attacker tokens accepted at each FPR. $\bar{Q}$ is the mean number of sampled victim responses per forged text. OriginMark uses a pool of $K = 8$ keys; the three baselines use a single fixed key.

Watermark	Qwen3.5-9B					Llama3.1-8B				
	TPR @ FPR			Median p-value	$\bar{Q}$	TPR @ FPR			Median p-value	$\bar{Q}$
	1%	0.1%	0.01%			1%	0.1%	0.01%		
Qwen3.5-2B										
+ MCMark	88%	72%	54%	4.6×10^{-5}	2,577	55%	36%	20%	6.8×10^{-3}	1,787
+ SynthID-Text	100%	96%	83%	1.2×10^{-6}	2,549	87%	68%	50%	9.5×10^{-5}	1,761
+ TextSeal	100%	98%	95%	1.4×10^{-7}	2,538	94%	79%	61%	1.8×10^{-5}	1,779
+ OriginMark	**0%**	**0%**	**0%**	**0.724**	2,562	**1%**	**0%**	**0%**	**0.553**	1,808
Llama3.2-3B										
+ MCMark	35%	18%	6%	0.022	1,588	60%	35%	13%	7.3×10^{-3}	2,010
+ SynthID-Text	60%	32%	13%	5.3×10^{-3}	1,591	83%	63%	44%	2.1×10^{-4}	2,052
+ TextSeal	72%	44%	29%	1.9×10^{-3}	1,593	96%	82%	62%	2.9×10^{-5}	2,010
+ OriginMark	**1%**	**0%**	**0%**	**0.863**	1,605	**1%**	**0%**	**0%**	**0.709**	2,009

placing 10% of completion tokens with different non-special vocabulary tokens, and using Gemma-4-31B to paraphrase one uniformly selected contiguous span covering 30% of the completion. The same pool-scanning detectors are applied after modification.

Random token replacement. Table 13 shows that distributed token replacements substantially weaken every watermark. At 400 tokens and 1% FPR, OriginMark retains 44.1% TPR, compared with 21.4%–31.9% for the 128-key baselines. At 200 tokens, the corresponding rates are 20.7% and 7.4%–14.2%. Its advantage over the strongest randomized baseline is therefore 12.2 and 6.5 percentage points, respectively. OriginMark leads these baselines at all reported length–threshold combinations except 200 tokens at 0.01% FPR, where TextSeal obtains 4.1% versus 4.0%.

Span paraphrasing. Table 14 reports the paraphrasing results. At 1% FPR, OriginMark achieves 29.4% TPR at 200 tokens and 58.1% at 400 tokens, exceeding the strongest 128-key baseline by

Table 11: Detection performance of **Llama-3.1-8B-Instruct** at different completion lengths and false-positive rates. TPRs are obtained by directly thresholding the Bonferroni-corrected p-values. Completions shorter than the target length are excluded, leaving 602–645 watermarked and 624 unwatermarked responses at 200 tokens and 113–145 watermarked and 154 unwatermarked responses at 400 tokens.

Watermark	Key Pool Size	200 Tokens				400 Tokens			
		TPR @ FPR			Median p-value	TPR @ FPR			Median p-value
		1%	0.1%	0.01%		1%	0.1%	0.01%	
Strong but Vulnerable									
MCMark	1	88.0%	78.0%	66.0%	1.36×10^{-6}	97.1%	92.9%	88.6%	2.36×10^{-19}
SynthID-Text	1	**95.6%**	**89.2%**	**79.4%**	5.49×10^{-8}	99.1%	**97.4%**	93.1%	$\mathbf{4.06\times10^{-27}}$
TextSeal	1	94.0%	85.8%	76.3%	$\mathbf{5.42\times10^{-8}}$	**99.3%**	96.4%	**93.4%**	2.75×10^{-18}
Robust to Attacks									
MCMark	128	70.4%	62.9%	54.1%	5.31×10^{-5}	86.9%	82.8%	77.9%	2.38×10^{-15}
SynthID-Text	128	75.2%	64.8%	56.5%	1.04×10^{-5}	92.6%	**90.4%**	84.4%	1.16×10^{-16}
TextSeal	128	80.8%	72.3%	63.1%	$\mathbf{2.87\times10^{-7}}$	92.5%	89.6%	86.6%	1.45×10^{-17}
OriginMark	8	**84.9%**	**74.6%**	**64.5%**	1.12×10^{-6}	**96.1%**	89.8%	**89.0%**	$\mathbf{6.77\times10^{-18}}$

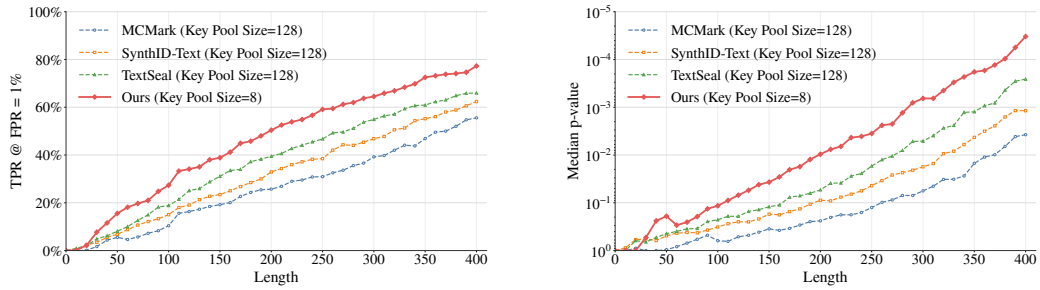
Table 12: Detection on **Gemma-4-31B** with C4 RealNewsLike prompts. TPRs use FPR thresholds on Bonferroni-corrected p-values. The evaluated settings retain 919–942 watermarked and 917 unwatermarked responses at 200 tokens, and 724–758 watermarked and 737 unwatermarked responses at 400 tokens. Bold denotes the best value within each group.

Watermark	Key Pool Size	200 Tokens				400 Tokens			
		TPR @ FPR			Median p-value	TPR @ FPR			Median p-value
		1%	0.1%	0.01%		1%	0.1%	0.01%	
Strong but Vulnerable									
MCMark	1	14.9%	4.7%	1.7%	0.180	26.6%	12.1%	6.2%	0.0743
SynthID-Text	1	11.8%	3.9%	1.2%	0.128	25.0%	10.2%	4.4%	0.0623
TextSeal	1	**20.2%**	**7.4%**	**2.5%**	**0.0982**	**35.6%**	**18.2%**	**9.8%**	**0.0369**
Robust to Attacks									
MCMark	128	1.9%	0.5%	0.1%	1.000	4.6%	1.2%	0.3%	0.700
SynthID-Text	128	2.3%	0.5%	0.1%	0.723	5.9%	2.0%	0.7%	0.559
TextSeal	128	3.8%	1.4%	0.4%	0.585	8.5%	3.8%	2.1%	0.434
OriginMark	8	**6.5%**	**2.4%**	**1.0%**	**0.440**	**14.2%**	**5.9%**	**2.4%**	**0.234**

8.6 and 15.4 percentage points. It also leads these baselines at both stricter thresholds. Under the tested settings, paraphrasing one 30% span leaves more detectable evidence than replacing 10% of tokens throughout the text. This is consistent with the locality of context-dependent watermarking: an isolated replacement can affect both its own score and the next h context-derived keys, whereas a contiguous edit leaves long untouched regions. The two modifications have different effects on meaning and fluency, so their percentages are not equivalent measures of attack severity. Fixed-key baselines retain stronger detection in these experiments, but remain vulnerable to the identification and spoofing attacks studied above.

F.5 Quality Evaluation of OriginMark

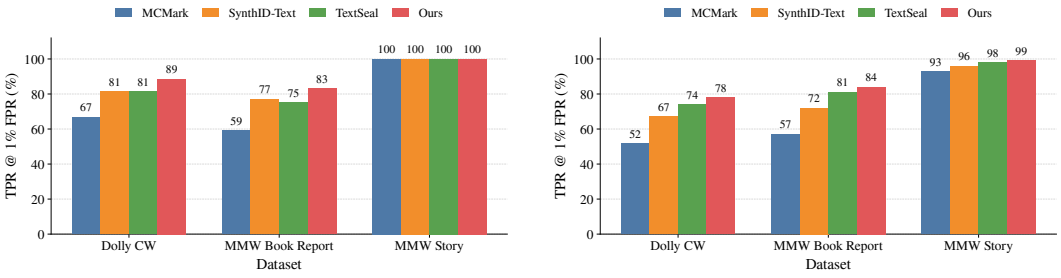
Table 15 compares Qwen3.6-27B with and without OriginMark on ten knowledge, STEM, and reasoning benchmarks. The mean score over all tasks is 85.7 without watermarking and 86.0 with OriginMark. The knowledge average changes from 84.2 to 84.3, and the STEM and reasoning



(a) TPR at a 1% FPR.

(b) Median pool-level p-value.

Figure 4: Detection versus completion length on Qwen3.8-27B using C4 RealNewsLike prompts. Baselines use 128 independent keys and OriginMark uses eight complementary keys. The detector scores the first L tokens of responses with at least L tokens and applies the Bonferroni correction. The zero-token point denotes no evidence (TPR $= 0$, $p = 1$).



(a) Qwen3.8-27B.

(b) Llama-3.1-8B-Instruct.

Figure 5: Detection across text domains at 200 tokens and a 1% FPR, using 128 independent keys for the baselines and eight complementary keys for OriginMark. Shorter completions are excluded: Dolly retains 86–87 responses per method for Qwen and 83–86 for Llama; each model retains 100 book reports and 96 stories per method. Bar labels are rounded to the nearest percentage point.

average from 86.7 to 87.2. Individual changes go in both directions: GPQA Diamond decreases by 2.5 points, while HMMT February 2026 increases by 4.6 points. The aggregate results show no observed degradation in this evaluation.

F.6 Time Cost Analysis

Attack query budget. We report cost in sampled victim responses, which does not depend on the hardware or batching policy. For identification, the 9 tested distances, $N = 100$ probes per distance, two prefixes per probe, and $R = 20$ samples per prefix require $9 \times 100 \times 2 \times 20 = 36{,}000$ responses per model and watermark configuration. The larger key-pool sweep uses $N = 1{,}000$, increasing this budget to 360,000 responses.

For spoofing, let B be the number of candidates, P the number of compared candidate pairs, m the samples per ordered pair, and T_{probe} the number of attacked positions. The response budget is

$$Q_{\text{spoof}} = 2mP\,T_{\text{probe}}. \tag{18}$$

Comparing all $P = \binom{B}{2} = 28$ pairs with $B = 8$ and $m = 100$ costs 5,600 samples per attacked position, or 1,120,000 samples for a 200-token completion when every position is attacked. The spoofing results in Tables 5, 9, and 10 use a reduced budget built on three observations. First, a single sample per ordered pair ($m = 1$) is sufficient, because the attacker needs only a ranking of the candidates and the log-count scores are pooled over both orders. Second, not all pairs are needed. At each attacked position, we pair every candidate with two random others, drawing a uniformly random 2-regular graph on the B candidates. This gives $P = B = 8$ distinct pairs and $2mP = 16$

Table 13: Detection performance of **Qwen3.8-27B** after uniformly replacing 10% of completion tokens with different non-special vocabulary tokens. TPRs are obtained by directly thresholding the Bonferroni-corrected p-values.

Watermark	Key Pool Size	200 Tokens				400 Tokens			
		TPR @ FPR			Median p-value	TPR @ FPR			Median p-value
		1%	0.1%	0.01%		1%	0.1%	0.01%	
Strong but Vulnerable									
MCMark	1	35.2%	**18.0%**	8.8%	3.80×10^{-2}	56.7%	39.7%	26.1%	4.63×10^{-3}
SynthID-Text	1	33.0%	16.0%	7.5%	4.30×10^{-2}	53.9%	33.6%	20.8%	7.98×10^{-3}
TextSeal	1	**36.8%**	17.8%	**10.7%**	$\mathbf{2.51\times10^{-2}}$	**61.4%**	**41.2%**	**27.1%**	$\mathbf{3.18\times10^{-3}}$
Robust to Attacks									
MCMark	128	7.4%	4.1%	2.2%	5.69×10^{-1}	21.4%	12.9%	9.6%	2.78×10^{-1}
SynthID-Text	128	7.7%	3.6%	2.3%	5.09×10^{-1}	21.9%	14.0%	8.8%	2.46×10^{-1}
TextSeal	128	14.2%	6.6%	**4.1%**	3.26×10^{-1}	31.9%	23.7%	15.4%	1.20×10^{-1}
OriginMark	8	**20.7%**	**8.1%**	4.0%	$\mathbf{2.21\times10^{-1}}$	**44.1%**	**27.5%**	**17.4%**	$\mathbf{2.50\times10^{-2}}$

Table 14: Detection performance of **Qwen3.8-27B** after a **30% span paraphrase by gemma-4-31B-it**, at different completion lengths and false-positive rates. One uniformly random contiguous span covering 30% of the completion tokens is paraphrased; tokens outside the span are preserved exactly. TPRs are obtained by directly thresholding the Bonferroni-corrected p-values.

Watermark	Key Pool Size	200 Tokens				400 Tokens			
		TPR @ FPR			Median p-value	TPR @ FPR			Median p-value
		1%	0.1%	0.01%		1%	0.1%	0.01%	
Strong but Vulnerable									
MCMark	1	44.4%	27.5%	16.9%	1.75×10^{-2}	70.2%	55.2%	41.1%	5.54×10^{-4}
SynthID-Text	1	42.7%	24.8%	14.5%	1.72×10^{-2}	72.8%	53.7%	37.8%	5.31×10^{-4}
TextSeal	1	**53.2%**	**30.7%**	**18.5%**	$\mathbf{7.27\times10^{-3}}$	**78.6%**	**60.2%**	**44.7%**	$\mathbf{2.56\times10^{-4}}$
Robust to Attacks									
MCMark	128	13.4%	7.4%	5.1%	4.54×10^{-1}	34.7%	25.1%	19.5%	8.97×10^{-2}
SynthID-Text	128	15.1%	9.7%	5.9%	3.65×10^{-1}	37.2%	26.3%	19.6%	6.31×10^{-2}
TextSeal	128	20.8%	13.2%	9.0%	2.46×10^{-1}	42.7%	33.5%	27.9%	2.58×10^{-2}
OriginMark	8	**29.4%**	**15.8%**	**9.7%**	$\mathbf{7.59\times10^{-2}}$	**58.1%**	**43.8%**	**33.7%**	$\mathbf{3.25\times10^{-3}}$

victim samples per attacked position. Each candidate's score is its add-one-smoothed log-count ratio against its two opponents, and the candidates receive the same rank bonuses as in the full protocol. Third, many positions carry little watermark signal or are not watermarked at all. We skip positions whose proxy entropy over its top-8 candidates is below 0.1 nats, as well as positions whose 4-token victim context already occurred in the text, since the watermark ignores repeated contexts. Skipped positions are sampled from the unmodified proxy distribution and issue no queries. When the attacker and victim use different tokenizers, the attacker only attacks whole-word positions, which it compares through $c \leq 8$ eligible candidates. Such positions cost $2c$ samples, and the rest issue no queries.

We generate every forgery with 512 tokens and the end-of-sequence token suppressed. The 200-token results use the first 200 attacker tokens, which coincide with a run stopped at 200 tokens, and $\bar{Q}$ counts only the queries issued at those positions. Out of 200 positions, the attacker attacks 104–178 on average. It skips 21–75 because of low entropy, fewer than 5 because of repeated contexts, and, for mismatched tokenizers, 28–50 because they are not whole-word positions. The resulting mean budget is $\bar{Q} = 1{,}570$–$2{,}857$ sampled responses per forged text, about 400–700 times smaller

Table 15: Downstream benchmark scores (%) for Qwen3.6-27B with and without OriginMark. Higher is better. Category averages are unweighted means over their tasks; the overall average weights each of the ten tasks equally.

		No Watermark	OriginMark
Knowledge	MMLU-Pro	85.9	85.6
	MMLU-Redux	93.8	94.2
	SuperGPQA	66.3	65.8
	C-Eval	90.9	91.4
	Avg	**84.2**	**84.3**
STEM & Reasoning	AIME26	94.2	96.7
	HumanEval	97.0	95.1
	GPQA Diamond	88.9	86.4
	HMMT Feb 25	86.7	85.8
	HMMT Nov 25	84.2	85.8
	HMMT Feb 26	68.9	73.5
	Avg	**86.7**	**87.2**
Overall Avg		**85.7**	**86.0**

than the full protocol, while still reaching 99%–100% acceptance at a 1% FPR for the Qwen3.5-2B attacker against Qwen3.5-9B.

Watermark generation and detection. At generation time, each response uses one selected key, so a pool of K keys does not require K separate model generations. Detection scans all K keys without receiving the generating key index. For an implementation that scores each key separately, the work is linear in K for fixed text length. OriginMark therefore requires eight key hypotheses per text, compared with 128 for the randomized baselines, a 16-fold reduction in the number of hypotheses evaluated.

G Possible Mitigation Methods

The attacks in Section 3.4 exploit token preferences that recur when a watermark configuration is reused. Besides the complementary key pool of OriginMark, possible defenses include rejecting texts that carry several key signatures and changing how token-level watermark randomness is derived. These approaches address different parts of the attack and introduce different detection costs.

G.1 Multiple-Key Detection

Aremu et al. (2025) propose sampling one key per response and accepting a text only when exactly one key is detected. The motivation is that an attacker learning from a mixture of keys may reproduce several signatures, whereas a genuine response uses one key. To evaluate this decision rule with our existing detectors, let

$$D_\alpha(x) = \sum_{j=1}^{K} \mathbb{1}\{p_j(x) < \alpha/K\}. \tag{19}$$

We accept if $D_\alpha(x) = 1$, reject as unwatermarked if $D_\alpha(x) = 0$, and flag a possible forgery if $D_\alpha(x) \geq 2$. Table 16 shows that accepting exactly one detected key still leaves 29.2%–36.5% acceptance at $K = 16$. At $K = 32$, nearly all accepted forgeries already trigger only one key. A mixture of weak signatures need not cause several detectors to cross their thresholds simultaneously. The rule therefore provides useful additional filtering, but does not eliminate spoofing in this setting. It also retains the cost of scanning the key pool and can reject genuine responses when a non-generating key crosses the threshold by chance.

Table 16: Spoofing acceptance (%) at a target global FPR of 1%, using the same per-key threshold $0.01/K$ for both rules. "Any" accepts one or more detected keys; "One" accepts exactly one. Lower is better. Each entry uses 96 saved forgeries.

	$K = 16$		$K = 32$	
Watermark	Any	One	Any	One
MCMark	50.0	36.5	13.5	13.5
SynthID-Text	47.9	33.3	11.5	10.4
TextSeal	47.9	29.2	12.5	12.5

G.2 Alternatives to Context-Derived Keys

Our probes preserve the last h tokens to reproduce the target watermark randomness. Removing this dependence would disrupt that particular mechanism. Context-derived keys nevertheless offer a practical benefit: a detector can reconstruct the watermark randomness from the text alone, and synchronization resumes once an intact context window is encountered after a local edit. Simply reusing one token-level key everywhere would instead expose the same preferences across unrelated contexts. Two alternatives illustrate the trade-offs.

Position-dependent keys. One can derive $k_t = H(sk, t)$ from the token index. A detector that assumes the original positions loses synchronization after an insertion, deletion, or removal of a prefix. Alignment search can address this issue, as in the robust distortion-free framework of Kuditipudi et al. (2024), but introduces additional computation and requires calibration of the search statistic. Moreover, resetting the same positional schedule for every response still reuses watermark randomness: an attacker could attempt probes at a fixed output position instead of a fixed context. Position dependence therefore changes the attack surface without by itself establishing resistance to repeated queries.

Cyclic key sequences. A finite ring of token-level keys can be traversed from a randomly chosen offset, as in the random-shift construction of Kuditipudi et al. (2024). This increases the variety of watermark schedules, but a ring of length L has only L starting offsets and eventually reuses its randomness. Detection must accommodate the unknown offset and, for edited text, possible misalignment. A longer ring can reduce reuse, while increasing the search space; the resulting detection power depends on the statistic and its null calibration. These alternatives merit evaluation, but changing the key schedule alone does not ensure the exact cancellation of token preferences required by Eq. 7.

H Baselines of Spoofing Attacks

Existing spoofing attacks fall into two families. *Frequency-based* attacks estimate the watermark rule by comparing token statistics in watermarked and unwatermarked text (Jovanovic et al., 2024). *Learning-based* attacks instead train a student on watermarked outputs and rely on the student to inherit the watermark signal (Gu et al., 2024; Sander et al., 2024; An et al., 2026). We evaluate representative attacks from both families against the fixed-key baselines of Section 5.1. We use the deployed configuration throughout: a context width of $h = 4$, an order-sensitive hash of the full context, and repeated contexts skipped. Each attack either could not forge text at all in this setting or produced a signal that did not exceed the unwatermarked control. We therefore evaluate spoofing in the main text with the probing attack of Section 3.4.

H.1 Frequency-Based Attacks

Watermark Stealing (Jovanovic et al., 2024) queries the victim for a large corpus of watermarked completions and generates a corpus of the same size with an unwatermarked auxiliary model. For each observed context and next token, the attack compares how often the pair appears in the two corpora. Pairs that are more frequent in the watermarked corpus are then boosted during forged

Table 17: Coverage of the candidate 5-grams (four-token context plus candidate token) encountered by Watermark Stealing. We generate 100 Dolly responses with Mistral-7B-v0.3 and pair each non-repeated context with the forger's top-20 candidates. A 5-gram is *overlapped* if it appears both in the 30,000 watermarked Qwen3.5-9B completions and in the 30,000 auxiliary completions. Only overlapped 5-grams give the attack a frequency ratio. Counts use the Mistral tokenizer.

	Total 5-gram	Overlapped 5-gram (percent)	Total Unique 5-gram	Overlapped Unique 5-gram (percent)
MCMark	616,840	3,337 (0.541%)	613,299	2,334 (0.381%)
SynthID-Text	616,840	3,316 (0.538%)	613,299	2,300 (0.375%)
TextSeal	616,840	3,215 (0.521%)	613,299	2,249 (0.367%)

Table 18: Watermark Stealing against context-width-4 watermarks with exact hashing. The victim is Qwen3.5-9B, and the auxiliary and forging model is Mistral-7B-v0.3. Results are for 100 Dolly prompts with up to 512 generated tokens. *Clean* is unmodified Mistral-7B-v0.3 output, and *Forged* uses the frequency-ratio reweighting. TPR is the fraction of responses with $p < 0.01$ under the victim's detector.

Watermark	Clean		Forged	
	TPR@1%↑	Median p-value↓	TPR@1%↑	Median p-value↓
MCMark	1%	0.591	0%	0.514
SynthID-Text	0%	0.428	1%	0.618
TextSeal	0%	0.566	1%	0.500

generation. The attack succeeds against the schemes it evaluated because their watermark keys depend on at most three context tokens. For its longest-context scheme, KGW2-SelfHash, the key also depends on the context only through the token of minimum hash. Statistics collected under many different contexts can therefore be pooled, and a modest corpus covers most keys. With an order-sensitive hash of four context tokens, the key changes whenever any context token changes. The attacker must then observe each exact 5-gram, meaning the four context tokens and the candidate token, in both corpora, or it has no evidence about that key.

We reproduce the attack with Qwen3.5-9B as the victim and Mistral-7B-v0.3 as the auxiliary and forging model. The victim answers 30,000 C4 RealNewsLike prompts with up to 512 tokens per completion under each watermark. The auxiliary model answers another 30,000 prompts. Following Jovanovic et al. (2024), forgery is evaluated on 100 Dolly prompts. Table 17 measures how much evidence the attack can use. During forged generation, the 100 responses contain 30,842 positions whose four-token context has not already appeared in the same response. We pair each of these contexts with each of the forger's top-20 candidates, giving 616,840 candidate 5-grams. Only about 0.5% of them appear in both corpora, and the fraction is nearly identical across the three watermarks. For the remaining 99.5%, the frequency ratio is undefined, and the attack falls back to the forger's own distribution.

We confirm this directly by running the forgery. For each exact context and candidate token, we compute the ratio of the pair's count in the watermarked corpus to its count in the auxiliary corpus, setting the ratio to one when either count is zero. We then sample from the forger's distribution reweighted by this ratio. The forged responses are retokenized with the victim tokenizer and passed to the victim's detector. Table 18 shows that the forged text is detected no more often than clean Mistral-7B text. In both cases, at most one of 100 responses is accepted at a 1% FPR, and median p-values remain between 0.4 and 0.7.

H.2 Learning-Based Attacks

A second family of attacks exploits *watermark radioactivity* (Sander et al., 2024). After repeated exposure to watermarked outputs, a student model may absorb the watermark's statistical patterns

Table 19: Supervised fine-tuning attack. The Qwen3.5-2B student is fine-tuned on 20,000 watermarked Qwen3.5-9B completions and evaluated with the teacher's detector on held-out responses that reach 512 tokens. The control is the same student trained on unwatermarked completions. Both students stay at the null level.

Watermark	Trained on watermarked text		Control (unwatermarked text)	
	TPR@1%↑	Median p-value↓	TPR@1%↑	Median p-value↓
MCMark	0.79%	0.516	0.19%	0.555
SynthID-Text	0.63%	0.536	0.45%	0.542
TextSeal	1.28%	0.437	0.90%	0.453

and then generate text that the victim's detector accepts. Prior work shows that such watermarks can be learned through distillation (Gu et al., 2024). We evaluate two variants.

Supervised fine-tuning. We fine-tune a student directly on watermarked teacher outputs. The teacher is Qwen3.5-9B, which answers 20,000 C4 RealNewsLike prompts with 512 tokens under each watermark. The student is Qwen3.5-2B, which we fine-tune on these completions with QLoRA (rank 64, $\alpha = 128$, learning rate 2×10^{-4}, effective batch size 128, one epoch). As a control, we train a second student in the same way on unwatermarked teacher completions. Both students answer 5,000 held-out prompts, and each response is scored by the teacher's detector with the teacher's key. Table 19 reports responses that reach the 512-token limit, between 1,489 and 1,639 per watermark. At a 1% FPR, the students trained on watermarked text are detected in only 0.63%–1.28% of responses. The control student is detected in 0.19%–0.90% of responses, and median p-values are similar in both cases (0.44–0.54 versus 0.45–0.56). Including all 5,000 responses gives the same picture. Watermarked-text students then reach 0.56%–1.04%, compared with 0.42%–0.82% for the control. The student therefore inherits no signal that separates it from the control. This contrasts with reported radioactivity for watermarks whose keys depend on shorter contexts (Sander et al., 2024; Gu et al., 2024).

DITTO. DITTO (An et al., 2026) distills a student from a watermarked teacher and extracts a watermark signature through next-token frequency comparisons conditioned on prefixes of at most two tokens. This signature is injected into an attacker model's logits. We evaluate the authors' implementation and released configuration on their two Llama teacher–student pairs, comparing the original SynthID-Text context width $h = 1$ (`ngram_len` $= 2$) with the official default $h = 4$ (`ngram_len` $= 5$). The attacker without signature injection serves as a control.

As shown in Table 20, DITTO achieves detection rates comparable to the teacher's at $h = 1$, whereas median p-values remain near the null median at $h = 4$ (0.51–0.56). The elevated TPR at $h = 4$ reflects detector miscalibration and repetition: the reference detector assumes independence across all 30 tournament layers, yielding smaller p-values for longer responses under the null, while spoofed responses are approximately twice as long as controls and contain 83% repeated word trigrams, compared with 10%–20% for controls. At a fixed length of 200 tokens, DITTO recovers 76% of the teacher's watermark score displacement at $h = 1$ for the Llama-3.2-3B→1B pair, but at most 12.5% at $h = 4$ across injection strengths 2.5–10. This limitation is consistent with the mismatch between DITTO's short-prefix statistics and SynthID-Text's exact four-token key dependence: four-token prefixes satisfy DITTO's frequency threshold at fewer than 7% of positions.

I Ablation Study

We vary the number of complementary channels in OriginMark and the number of independent keys in the baselines to examine detection and attack resistance.

I.1 Complementary Pool Size

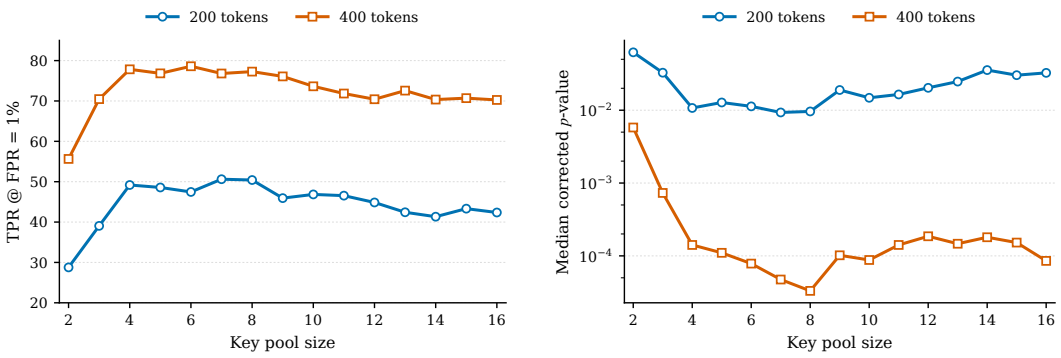
We evaluate OriginMark on Qwen3.8-27B with C4 RealNewsLike prompts, varying K from 2 to 16 with a separately calibrated partition for each pool. Figure 6 shows a broad detection peak around

Table 20: DITTO against SynthID-Text with the authors' implementation, teacher–student pairs, and detection script. Results are for 100 held-out Dolly prompts with up to 512 generated tokens and injection strength $\alpha = 4.5$. *Control* is the attacker model without the injected signature. Rep. is the median fraction of repeated word trigrams per response. At context width $h = 4$, the median p-value of DITTO stays at the null level, and its residual TPR comes from response length and repetition (see text).

h	Teacher→Student	Watermarked teacher		DITTO			Control		
		TPR	Median p	TPR	Median p	Rep.	TPR	Median p	Rep.
1	Llama3 8B→3B	98%	3.4×10^{-16}	93%	2.0×10^{-18}	0.90	6%	0.645	0.20
	Llama3 3B→1B	86%	1.7×10^{-14}	86%	1.1×10^{-14}	0.92	5%	0.459	0.12
4	Llama3 8B→3B	94%	2.8×10^{-9}	31%	0.506	0.83	3%	0.669	0.20
	Llama3 3B→1B	82%	5.6×10^{-7}	26%	0.561	0.83	2%	0.604	0.10

$K = 4$–8. At 400 tokens and a 1% FPR, TPR rises from 55.6% at $K = 2$ to a maximum of 78.6% at $K = 6$, then falls to 70.3% at $K = 16$. The 200-token results follow a similar trend.

Larger pools lower the null hit probability $1/K$, but increase the Bonferroni penalty and can leave more channels with little probability mass (Eq. 16). We choose $K = 8$: its 400-token TPR (77.3%) is near the observed maximum, and its median corrected p-value (3.31×10^{-5}) is the lowest in the sweep. These results support a practical choice rather than a unique optimum.



(a) TPR at a 1% FPR.

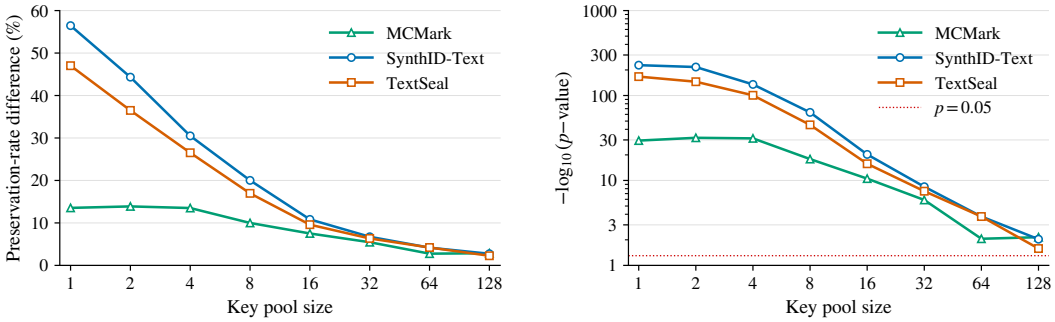
(b) Median Bonferroni-corrected p-value.

Figure 6: OriginMark detection versus complementary pool size on Qwen3.8-27B with C4 RealNewsLike prompts. Each pool uses a separately calibrated partition. Detection scores the first 200 or 400 tokens, excludes shorter completions, and applies the Bonferroni correction.

I.2 Independent Key Pools and Watermark Identification

We next vary the baseline pools over $K \in \{1, 2, 4, 8, 16, 32, 64, 128\}$ on Qwen3.5-9B, using 1,000 probes per perturbation distance. Figure 7 shows that larger pools generally reduce the preference-retention gap across the context boundary. From $K = 1$ to 128, this gap falls from 13.54 to 2.87 percentage points for MCMark, 56.46 to 2.74 for SynthID-Text, and 47.02 to 2.27 for TextSeal, approaching the clean model's 2.65-point gap.

The clean model itself yields $p = 0.0115$, so statistical significance alone does not establish watermark deployment. Mitigation at $K = 128$ is supported by the smaller effect relative to the clean reference. The plotted Fisher p-values test a specified boundary and are not the boundary-search-adjusted p_{id}.



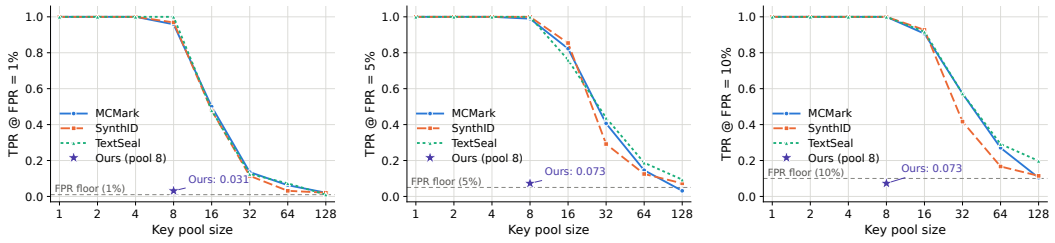
(a) Preference-retention difference.

(b) Raw Fisher test significance.

Figure 7: Identification versus independent pool size on Qwen3.5-9B ($h = 4$). Left: retention at distances 5–10 minus that at 2–4, in percentage points. Right: raw one-sided Fisher test $-\log_{10}(p)$; the dotted line marks $p = 0.05$.

I.3 Independent Key Pools and Spoofing

Figure 8 evaluates a Qwen3.5-2B attacker against a Qwen3.5-9B victim on 96 MMW-story prompts, with 1,024 generated tokens per prompt. At the 1% threshold, baseline acceptance falls from 95.8%–100% at $K = 8$ to 1.0%–2.1% at $K = 128$. OriginMark accepts $3/96$ forgeries (3.1%) with only eight keys, and $7/96$ (7.3%) at both the 5% and 10% thresholds. Residual vulnerability remains: TextSeal accepts 19.8% at $K = 128$ under the 10% threshold. Thus, larger independent pools suppress the tested attacks but incur a detection cost: at 400 tokens and 1% FPR, their TPRs are 55.6%–66.0%, compared with 77.3% for OriginMark (Table 6).



(a) FPR 1%.

(b) FPR 5%.

(c) FPR 10%.

Figure 8: Spoofing acceptance under the attack versus independent pool size (lower is better). The OriginMark reference uses $K = 8$ throughout. Dashed gray lines mark FPRs. Detectors use Bonferroni-corrected p-values.