REDIFFMARK: POST-HOC TEXT WATERMARKING BY LOCAL RE-DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-hoc watermarking embeds detectable signals into existing text, a capability crucial for closed-API outputs, RAG-retrieved passages, and published documents. However, such methods must minimally alter the original text while remaining computationally efficient for large-scale deployment. Existing sparse-editing schemes preserve the text but suffer from poor signal stability and low efficiency due to rule-based byte signaling; conversely, full-rewriting schemes carry strong signals but rewrite most tokens, leading to significant text alteration and slow speeds. We observe that discrete diffusion language models (dLLMs) can bridge this gap. To this end, we propose ReDiffMark, a watermarking paradigm based on localized re-diffusion for text that is stored or served verbatim or with light edits. We address challenges regarding available contextual entropy and the difficulty of identifying sparse edit locations through entropy-gated editing with key-seeded exponential-minimum (Gumbel) sampling, employing a replay-based, surprisal-weighted detector that re-runs the same dLLM on the received text under a calibrated null hypothesis. Extensive experiments across dLLM families, text sources, and passage lengths show that ReDiffMark attains high single-passage detection at low edit rates with a constant two-forward-pass cost.

1 INTRODUCTION

Text watermarking embeds an imperceptible, key-verifiable signal into text and has become a practical means of content authentication and provenance tracking (Liu et al., 2025). The need, however, does not stop at text produced under the watermarker's own decoder: outputs of closed APIs, passages retrieved into RAG pipelines, and already-published documents all arrive as *finished* text whose generation process is inaccessible. This motivates *post-hoc* watermarking, i.e. embedding a detectable signal into an arbitrary existing passage without knowing how it was produced. That the text is already finished also imposes tighter constraints than at generation time: the passage may contain facts and entities that must survive and cannot be rewritten wholesale for the sake of a watermark, attribution is usually needed on a single short passage rather than a large aggregate, and the cost must permit deployment at scale. A practical post-hoc watermark must therefore be strong on one passage, change little, and run cheaply, all at once.

Existing post-hoc methods fall into two families at the opposite ends of a trade-off (Figure 1). Sparse in-place schemes substitute a few words using synonym rules or masked-LM candidates (Yang et al., 2022; Yoo et al., 2023): they preserve the original text well, but each edit signals a single hard-coded bit, so the signal per passage is weak and unstable, and candidate construction is slow. Full-rewriting schemes instead regenerate the entire passage with an instruction-tuned autoregressive LLM under a generation-time watermark (Lau et al., 2024; Chang et al., 2024; Fernandez et al., 2026): they detect well, but keep almost nothing of the original verbatim, distort facts and entities, and their sequential decoding cost grows linearly with length. In short, the field offers "sparse but weak per edit" or "strong but invasive", and neither satisfies the three requirements of the post-hoc setting simultaneously.

The critical problem left open by both families is the *strength of a single edit*: no existing post-hoc scheme makes a small number of local, meaning-preserving edits each carry a strong, verifiable signal. This raises the question this paper answers: ***can a handful of local edits each carry a strong and verifiable watermark signal?***

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.

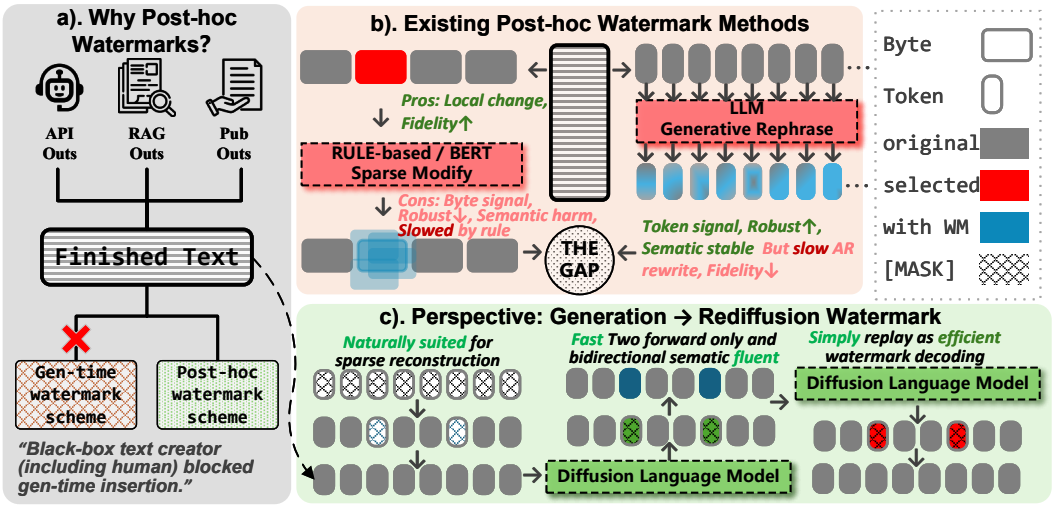


Figure 1: **The post-hoc setting and the gap ReDiffMark fills.** Left: finished text arrives without a decoder, so generation-time watermarking does not apply. Right: sparse rule-based substitution changes few tokens but carries one bit per edit, and full rephrasing detects well but rewrites the passage; ReDiffMark makes a few keyed re-diffusion edits, each carrying a continuous score, while keeping every other token verbatim.

Our idea is simple: *use a masked diffusion language model (dLLM) to locally re-diffuse the text*. Because a dLLM is trained at every mask rate, re-masking a few positions of an existing passage yields a valid late state of the reverse diffusion process, and resuming from that state is exactly generation-time watermarking applied to the last few denoising steps of a generation that was never actually run. Every property that so far belonged to generation-time watermarks, including per-sample distortion-free sampling and continuous per-token scores, is inherited directly at the re-masked positions, while every unmasked token remains verbatim, and the parallel bidirectional infilling costs a constant two forward passes at any length. To our knowledge, all existing dLLM watermarks act at generation time (Gloaguen et al., 2025; Wu et al., 2025; Yang et al., 2026; Hong & No, 2026); none uses a dLLM as a post-hoc editor of existing text. We instantiate this idea as ReDiffMark, a post-hoc watermark by *local re-diffusion*: a secret key selects a sparse set of positions, the dLLM refills them under the full bidirectional context with Gumbel sampling, and a detector that holds the key and the same dLLM, but no original text, decides based solely on the received passage.

Realizing this idea, however, is non-trivial: moving from autoregressive rewriting to local re-diffusion raises one technical challenge on each side of the channel.

- **Challenge 1: *Entropy compression*.** A generation-time watermark modulates a token whose continuation is still unwritten, whereas a post-hoc refill fills a hole whose left *and* right neighbours are fixed real text; the bidirectional conditional therefore collapses onto the original token, and roughly half of the selected positions carry too little entropy to hold any signal at any perplexity cost.
- **Challenge 2: *Receiver asymmetry*.** The detector holds neither the original text nor the identity of the positions that were actually edited, and the context it observes is no longer the context the embedder conditioned on; scoring positions that carry no signal dilutes the statistic as $\sqrt{T}$.

To tackle *entropy compression*, we introduce **entropy-gated modulation**. The key insight is that the scarce entropy must be neither wasted nor biased: an entropy-and-candidate gate spends the edit budget only on positions whose refill distribution is genuinely multi-modal *and* still contains the original token, so every edit is a substitution among near-equivalents rather than a correction; at those positions, key-seeded exponential-minimum (Gumbel) sampling embeds a *continuous* score per edit while remaining distortion-free, extracting far more evidence per position than any binary green-list rule can at this budget.

To tackle *receiver asymmetry*, we introduce **gated replay detection**. The key insight is that the detector can reconstruct from the received text and the key the quantities needed for scoring: it reruns

the same keyed selection and two-pass checkerboard schedule to obtain replayed refill distributions, then reapplies the entropy gate to restrict the statistic to positions likely to carry signal. A surprisal-weighted sum of the resulting Gumbel scores admits a closed-form, empirically calibrated conditional null for rigorous false-positive control.

In summary, our contributions are as follows:

1. We propose ReDiffMark, to our knowledge the first post-hoc text watermark that applies local re-diffusion with a masked diffusion language model to arbitrary existing text, combining sparse, verbatim-preserving edits with parallel bidirectional infilling at the cost of two forward passes.
2. We identify and characterize the two challenges specific to post-hoc re-diffusion: entropy compression and receiver asymmetry, and address them with entropy-gated modulation, a continuous distortion-free signal designed for this sparse channel, and gated replay detection, a surprisal-weighted detector with a closed-form, empirically calibrated conditional null.
3. We conduct an extensive empirical study showing that ReDiffMark achieves high single-passage detection at low edit rates across text sources and dLLM families, analyze its trade-offs, and demonstrate its extension to multi-bit payloads and multi-source provenance.

2 Related Work

Generation-time watermarks. Most LLM watermarks modify the decoder: green-list logit bias (Kirchenbauer et al., 2023; 2024; Zhao et al., 2024), unbiased reweighting (Hu et al., 2024; Wu et al., 2024), and distortion-free exponential-minimum or tournament sampling (Kuditipudi et al., 2024; Christ et al., 2024; Dathathri et al., 2024); entropy-aware variants score only high-entropy positions (Lee et al., 2024; Lu et al., 2024) , while multi-bit extension encodes the message through position allocation, message seed randomness, or error-correcting codes (Wang et al., 2024; Yoo et al., 2024; Fernandez et al., 2023; Qu et al., 2025; Kim et al., 2026; Park et al., 2026; Wang et al., 2026; Guo et al., 2026). All require control of sampling; we borrow the Gumbel statistic and the entropy gate but apply them to text that already exists. Watermarks for masked diffusion LMs (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2025; Nie et al., 2025; Zhu et al., 2026; Ye et al., 2025) likewise act at generation time (Gloaguen et al., 2025; Wu et al., 2025; Yang et al., 2026; Hong & No, 2026); none uses a dLLM as a post-hoc editor.

Post-hoc watermarks. Most generation-time watermarks cannot be directly applied to post-hoc settings because they require specific sampling procedures, but post-hoc text limits their generation. Early schemes substitute words in place with synonym sets or masked-LM infilling (Topkara et al., 2006; He et al., 2022a;b; Yang et al., 2022; Yoo et al., 2023; Zhang et al., 2024b; Munyer & Zhong, 2023). As the development of large language models, methods like rephrasing rewrite the passage with an instruction-tuned LM under a generation-time watermark (Fernandez et al., 2026; Lau et al., 2024; Bahri & Wieting, 2026). They control only the sampling of a black-box model; PostMark (Chang et al., 2024) selects key-dependent words and inserts via LM. ReDiffMark is sparse and in place like the first family, but replaces rules and small infillers with a bidirectional dLLM, a per-sample modulation and a replay detector with a calibrated continuous statistic. Paraphrasing, watermark stealing and impossibility results (Krishna et al., 2023; Sadasivan et al., 2025; Jovanovic et al., 2024; Zhang et al., 2024a) bound what any watermark achieves (Section 5.5, Section 6); per-source keys target the RAG attribution use case of Jovanovic et al. (2025); Tolstokorov et al. (2026).

3 Problem Setup

Setting and threat model. A watermarking party receives a finished text $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{V}^n$ of unknown provenance (human-written, closed API, etc.), together with a secret master key sk. It must output $\boldsymbol{x}'$ that keeps $x'_i = x_i$ outside a small edited set, stays fluent and faithful to $\boldsymbol{x}$, and lets a detector holding sk but *not* $\boldsymbol{x}$ decide at a calibrated false-positive rate whether a received passage $\boldsymbol{y}$ carries the watermark; latency must be fixed and small enough for inline use. Both parties hold sk and a masked diffusion LM $\mathcal{M}$ (a cheaper key-only mode that needs only the tokenizer, Section 4.5); everything they must agree on (filter, selection rate, schedule) is public, and all per-position randomness comes from sk through a keyed PRF, so an adversary without the key cannot predict which positions were edited or which tokens are favored, but may edit the text (Section 5.5).

Masked diffusion language models. A masked diffusion LM (dLLM) such as LLaDA (Nie et al., 2025) is a bidirectional Transformer trained to predict masked tokens under mask rates drawn from $[0, 1]$: when a set $\mathbb{S} \subseteq [n]$ of positions is replaced by `[MASK]`, one forward pass returns for every $i \in \mathbb{S}$ a categorical distribution

$$p_i(\cdot) = \mathcal{M}(\, x_i \mid \boldsymbol{x}_{[n]\setminus\mathbb{S}}\,), \tag{1}$$

conditioned on the *unmasked* tokens on both sides; the embedder samples each masked position from its own marginal, the standard parallel-decoding step of masked diffusion, and this per-position marginal is the only object the scheme uses.

Exponential-minimum (Gumbel) sampling. Given a distribution p over $\mathcal{V}$ and i.i.d. uniforms $u_v \sim \mathrm{Unif}(0, 1)$, $v \in \mathcal{V}$, the token

$$y = \arg\max_{v \in \mathcal{V}} \frac{\log u_v}{p_v} \tag{2}$$

is distributed exactly according to p, so sampling with key-seeded uniforms is *distortion-free per sample* (Kuditipudi et al., 2024). Conditioned on y, u_y has CDF u^{1/p_y}. Detector that recomputes u_y from the key scores $s_y = -\log(1 - u_y)$, which is $\mathrm{Exp}(1)$ under the null and has mean $\mathcal{H}_{1/p_y} \approx \log(1/p_y) + 0.577$ under the alternative. This fact weights the position at detection time (Section 4.5).

4 ReDiffMark: Watermarking by Local Re-Diffusion

4.1 Overview

ReDiffMark treats the dLLM as a keyed bidirectional infiller: it never generates text, it only re-decides a sparse, key-chosen subset of the tokens that are already there. Embedding maps a passage $\boldsymbol{x}$ and a key to a passage of the same length in four steps, whose scoring-relevant quantities the detector can reconstruct from the received text and the key (Algorithm 1, Appendix A.1): *Rule filter* (Section 4.2). A deterministic, text-only filter $\mathcal{R}(\boldsymbol{x})$ removes the positions that must never be touched: entities, numbers, sub-words and punctuation. *Keyed selection* (Section 4.2). A PRF of the key selects a ρ-fraction of the remaining positions; without the key the selected set is a uniformly random subsample. *Checkerboard re-diffusion* (Section 4.3). The selected positions are masked in two passes of alternating parity, one forward each, so every masked position is refilled with real tokens on both sides. *Gate and modulate* (Section 4.2, Section 4.4). A masked position is edited only if its refill distribution carries enough entropy and still contains the original token (Equation 3); the new token is then drawn from the top-p nucleus by key-seeded Gumbel sampling (Equation 4), which leaves that distribution undistorted.

4.2 Where to edit: Filter, selection, and model gate

Filter. $\mathcal{R}(\boldsymbol{x}) \subseteq [n]$ is computed from the token strings only: it excludes the first token, tokens with digits, punctuation-only tokens, sub-word continuations and the tokens preceding them , non-sentence-initial capitalized words, and tokens inside URLs, code spans and protected patterns (Appendix A).

Keyed selection. Let $\mathsf{k}_{\mathrm{sel}} = \mathrm{HMAC}(\mathsf{sk}, \texttt{"select"})$. Position i is selected if $\mathrm{PRF}(\mathsf{k}_{\mathrm{sel}}, i) < \rho$, with $\mathrm{PRF}(\cdot) \in [0, 1)$, giving the selected set

$$\mathcal{S} = \{i \in \mathcal{R}(\boldsymbol{x}) : \mathrm{PRF}(\mathsf{k}_{\mathrm{sel}}, i) < \rho\}.$$

$\mathcal{S}$ depends only on positions and the text-only filter, so the detector recomputes it from any received text of the same length while an adversary without sk sees a uniformly random ρ-subsample (multi-token selection in Appendix D.1).

Model gate. The mask position i can be edited using the infill distribution p_i (Equation 1) only if

$$\mathrm{H}(p_i) \;\geq\; h_{\min} \qquad \text{and} \qquad x_i \in \mathrm{TopK}(p_i, k) \cup \mathrm{Nucleus}(p_i, p_{\mathrm{gate}}), \tag{3}$$

i.e. the conditional is genuinely multi-modal *and* still contains the original token; positions that fail keep their token. The first condition is the entropy gate of SWEET (Lee et al., 2024), the second turns

every edit into a substitution among near-equivalents rather than a "correction". We use $h_{\min} = 0.5$ nats, $k = 20$, $p_{\text{gate}} = 0.95$, a hard cap of 25% edited tokens, and write $T_{\text{e}} = |\mathcal{S}_{\text{e}}|$ for the gated set.

The gate is a response to entropy compression rather than a restriction: half of the key-selected positions carry less than 0.5 nats and cannot carry a watermark signal at any perplexity cost, so the gate spends the budget only on the $T_{\text{e}} \approx 20$ of 41 positions that can (Figure 8).

4.3 Checkerboard re-diffusion

Positions masked in the same forward pass are each sampled from their own marginal given the unmasked context, so two adjacent selected positions masked together would be refilled without seeing each other and drift jointly (Section 5.7). ReDiffMark therefore masks the even-indexed selected positions in pass 0 and the odd-indexed ones in pass 1: within a pass every masked position sees real tokens on both sides, and the second pass conditions on the first. Each pass is a single forward over the whole sequence, so embedding costs two forward passes at any length and any ρ.

4.4 Distortion-free modulation with position-indexed keys

Let $\mathsf{k}_{\text{g}} = \text{HMAC}(\mathsf{sk}, \texttt{"global"})$. At a gated position we restrict p_i to its top-p nucleus (the smallest set with mass $\geq p_{\text{s}}$, $p_{\text{s}} = 0.9$ by default) and renormalise into q_i, then draw

$$y_i = \underset{v \in \text{supp}(q_i)}{\arg\max} \frac{\log u_{i,v}}{q_{i,v}}, \qquad u_{i,\cdot} = \text{PRF}(\mathsf{k}_{\text{g}}, \texttt{"gumbel"}, i) \in [0,1)^{|\mathcal{V}|}, \tag{4}$$

with the uniforms seeded by the key and the *absolute position* i. By Equation 2, $y_i \sim q_i$ exactly: the watermark is distortion-free relative to the truncated distribution, and truncation is a fidelity knob trading edit rate against perplexity (Section 5.2), not a requirement for calibration.

Key context Generation-time schemes seed the per-token randomness with the preceding token, but the post-hoc left neighbor may itself be a masked position whose value is not yet fixed, so we key on the absolute position by default. A left-context mode that keys both selection and uniforms on x_{i-1} is also supported; it is indistinguishable from position keys on clean text ($\text{TPR@1\%} = 0.913 \pm 0.012$) and differs only under insertions and deletions, which shift every downstream index for position keys but de-synchronize at most one position for left-context keys (Section 5.5, Appendix A.1).

4.5 Detection by replay with a surprisal-weighted statistic

The receiver is asymmetric: it holds neither the original text nor the identity of the positions that were actually changed, and most selected positions were left unedited, so every position scored without signal adds zero-mean noise that dilutes the statistic. The detector therefore scores only the positions the embedder could have edited, which it recovers by replaying the embedding, and uses a continuous per-position score, since a binary green-list count cannot separate a passage at this budget (Appendix B.3). Given a received text $\boldsymbol{y}$ and the key, the detector recomputes $\mathcal{R}(\boldsymbol{y})$ and $\mathcal{S}$, then *replays* the two checkerboard passes on $\boldsymbol{y}$: it masks the same positions with the same parity schedule and reads off, at every $i \in \mathcal{S}$, the replayed distribution $\hat{p}_i$ and the received token's probability $\hat{p}_i(y_i)$. The schedule and the per-position uniforms are recovered exactly, while the selection is recomputed from the received text using the same rule. The replayed $\hat{p}_i$ closely tracks the embedding-time distribution (exactly so for the second pass, whose context is the received text itself). It keeps the positions with $\text{H}(\hat{p}_i) \geq h_{\min}$ (the candidate-set gate is slightly harmful at detection and is omitted), recomputes $u_i = \text{PRF}(\mathsf{k}_{\text{g}}, \texttt{"gumbel"}, i)_{y_i}$, and forms

$$S(\boldsymbol{y}) = \sum_{i \in \hat{\mathcal{S}}_{\text{e}}} w_i \big(-\log(1 - u_i) \big), \qquad w_i = \text{clip}\big(-\log \hat{p}_i(y_i),\, 0,\, w_{\max} \big), \tag{5}$$

with $w_{\max} = 3$. The weights follow from the mean shift of Section 3: the expected score of an edited position exceeds its null value by $\mathcal{H}_{1/p_y} - 1$, so the locally optimal linear test weights each position by that shift, whose leading term $\log(1/p_y)$ we use (it retains 99.7% of the optimal efficiency on the replayed $\hat{p}$, Appendix B.1); the clip protects against the heavy tails of $-\log \hat{p}$ where the replay is imperfect.

Proposition 1 (Conditional null). *Under H_0 (the text was not produced with key k_g), the uniforms u_i are i.i.d.* $\mathrm{Unif}(0,1)$ *and independent of $\boldsymbol{y}$, so* $-\log(1-u_i) \overset{iid}{\sim} \mathrm{Exp}(1)$ *and, conditionally on the text, $S(\boldsymbol{y})$ is a weighted sum of independent* $\mathrm{Exp}(1)$ *variables with key-independent weights w_i: a hypo-exponential law, reducing to a scaled Gamma law when the weights are equal (and exactly* $\mathrm{Gamma}(T,1)$ *for unit weights) (Appendix B.1).*

For general weights we approximate this law by the moment-matched $\mathrm{Gamma}(\kappa,\theta)$, $\kappa = (\sum_i w_i)^2/\sum_i w_i^2$, $\theta = \sum_i w_i^2/\sum_i w_i$, and report $z = \Phi^{-1}(1-\Pr[\mathrm{Gamma}(\kappa,\theta) \geq S(\boldsymbol{y})])$; the exact hypo-exponential CDF can be evaluated numerically when needed. The approximation is tight in our regime (Figure 7), though every TPR in the paper is reported against *empirical* null thresholds, and detection costs the same two forward passes as embedding. A *key-only* detector that skips the replay ($w_i \equiv 1$, no gate) needs only the tokenizer but is $\approx 1.7\,z$ weaker (Section 5.6).

4.6 Extensions: multi-bit messages and multi-source provenance

The same key material carries payloads and attributes sources. A payload $\boldsymbol{b} \in \{0,1\}^m$ is carried by giving every gated position a symbol $s_i \in \{0,\ldots,M-1\}$ and seeding its Gumbel field with it, $u_{i,\cdot}^{(s)} = \mathrm{PRF}(\mathsf{k}_g, \texttt{"gumbel"}, i, s)$, which is distortion-free for every fixed symbol; a code maps $\boldsymbol{b}$ to the symbols, from one bit per position (Yoo et al., 2024) to the whole message as one symbol (Fernandez et al., 2023). The detector scores the received token under every symbol's field and decodes by maximum likelihood over the 2^m candidates with Equation 5, using the decoded message's score as the presence statistic, whose null is inflated by that maximum and calibrated empirically (Appendix D.7). For a document assembled from several sources, every span is watermarked with its source's key $\mathsf{k}_s = \mathrm{HMAC}(\mathsf{sk}, \texttt{"source"}, s)$ under a shared selection key, and the detector scores every candidate key at every scored position and runs Viterbi decoding over the states $\mathcal{C} \cup \{\varnothing\}$, which yields a per-source z and a token-level segmentation (Appendix D.8).

5 Experimental Evaluation

5.1 Setup

Data. The main benchmark is human-written news: 500 passages of C4 `realnewslike` (Raffel et al., 2020), each cut to 200 LLaDA tokens, with an empirical null of 1,000 disjoint unwatermarked passages scored under every key. Generality is tested on Wikipedia paragraphs, on continuations written by three instruction-tuned LLMs, on C4 passages of 100 to 500 tokens, on GSM8K solutions and MBPP programs, and on documents that alternate two sources. Throughout, n counts passages.

Models and operating points. The dLLM is LLaDA-8B-Base (Nie et al., 2025), with LLaDA-8B-Instruct and Dream-v0-Instruct-7B (Ye et al., 2025) as controls; the scoring and attack models are separate from it. Unless stated otherwise, experiments use the hyper-parameters of §4 at the *default point*, ρ=0.5 with p_s=0.9. The text-source generality study instead fixes ρ=0.35 with p_s=0.9 across all sources, while experiments that sweep ρ are marked explicitly. ReDiffMark numbers on C4 are means over five master keys ($\pm$ the across-key standard deviation), three for the left-context row and for every baseline; other experiments run under one key unless stated.

Metrics. Fidelity is measured against the original passage by the fraction of changed tokens, the relative increase in perplexity under an independent LM, sentence similarity and the fraction of entities kept verbatim. Detectability is the mean statistic z of Equation 5 and the true-positive rate at 10, 1 and 0.1% FPR, with thresholds read off the empirical null (Hong & No, 2026). Robustness is TPR@1% after an attack, read against a paired null in which the same attack is applied to the unwatermarked originals. Cost counts forward passes and token-forwards and measures seconds per passage; the composite Σ of Table 1 averages the four groups. Exact definitions and the abbreviations used in the tables are in Appendix A.2.

Baselines. We compare against post-hoc watermarks only, which receive finished text and control no decoder: the sparse substitution schemes ContextLS (Yang et al., 2022) and Yoo et al. (2023), the full read then insertion scheme PostMark (Chang et al., 2024), and TextSeal-style full rewriting (Sander

Table 1: **Post-hoc Watermarks Comparison.** 200-token C4 passages; mean ± across-key std; Sub. and BT are TPR@1% under 10% substitution and back-translation against a paired null; Σ is the unweighted composite of the four column groups (Table 5).

	Fidelity				Detectability		Robustness		Cost	
Method	Edit %	ΔPPL %	Sim.	Ent.	TPR@1%	TPR@.1%	Sub.	BT	s/p	Σ
Masked LM, sparse substitution										
ContextLS	9.26† ±0.31	18.6±2.2	0.996±.000	0.986±.004	0.580±.040	0.380‡ ±.092	0.107±.012	0.040±.020	91.0	37.0
Yoo et al.	2.13±0.27	19.5±3.6	0.998±.000	0.988±.001	0.898±.134	0.804±.236	0.103±.012	0.031±.012	13.4	53.1
Autoregressive LLM, insertion / full rewrite										
PostMark, r=0.06	65.1* ±0.4	27.6±1.3	0.924±.000	0.931±.003	0.954±.014	0.866±.057	0.779±.073	0.673±.058	5.9	69.4
TextSeal, KGW	78.1* ±0.5	434±617	0.886±.001	0.738±.006	0.949±.016	0.858±.033	0.853±.042	0.380±.060	9.9	57.4
TextSeal, Gumbel	77.8* ±0.7	153±114	0.877±.009	0.733±.017	0.996±.005	0.992±.007	0.987±.009	0.802±.045	2.0	73.6
Diffusion LM, local re-diffusion (ours)										
ReDiffMark, default	8.09±0.84	15.9±2.5	0.978±.003	0.991±.001	0.939±.018	0.874±.072	0.627±.062	0.056±.017	0.37	76.8
ReDiffMark, default, left-ctx	7.81±0.63	15.1±1.5	0.978±.001	0.988±.010	0.913±.012	0.827±.012	0.700±.067	0.338±.041	0.40	**80.0**

*edit distance; †against ContextLS's own detokenization; ‡null of 200 passages.

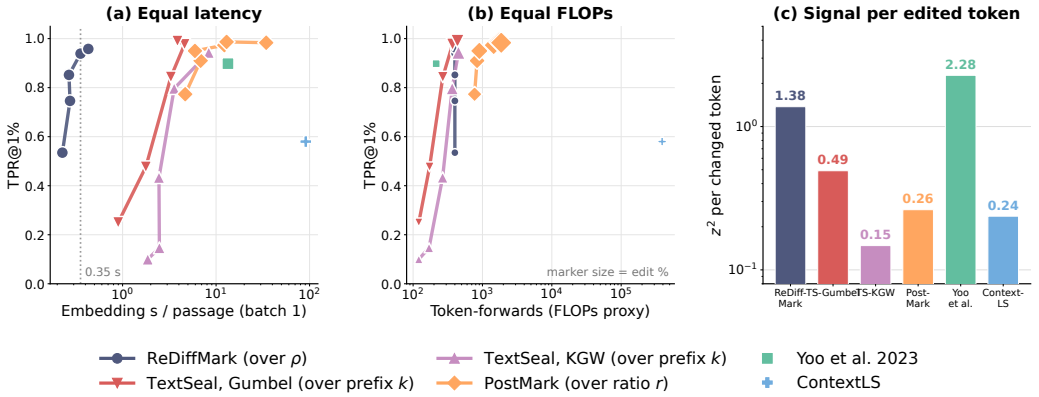


Figure 3: **Detection against its cost.** 200-token C4 passages; TPR@1% against embedding latency and against token-forwards, with the rewriting baselines throttled by the rewritten prefix length and PostMark by its insertion ratio, and z^2 per changed token.

et al., 2026) under KGW or Gumbel sampling. All run the authors' code on the same passages; their configuration and our zero-bit adaptation of the two substitution schemes are in Appendix A.3.

5.2 Detectability

Detection power follows the entropy the dLLM finds at the selected positions (Figure 2; Appendix D.2). We measure TPR@1% against mean z over passage lengths, text sources, dLLMs and domains. TPR@1% rises monotonically with ρ, from 0.54 at ρ=0.15 to 0.96 at 0.6, and the null FPR at $z \geq 4$ stays below 4×10^{-4} in every configuration (Appendix B.1). It reaches 0.75 on 100-token passages and 1.0 from 300 tokens; on Wikipedia and LLM-written text, which the dLLM predicts more easily, it falls to 0.58–0.77 at equal ρ; Dream-v0-Instruct-7B reproduces the default point (0.92); arithmetic and code expose almost no editable positions (§6).

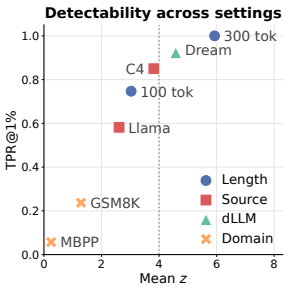


Figure 2: **Detection variety.**

5.3 Cost and Compute Efficiency

Two forward passes suffice at any length and any ρ (Figure 3). Embedding and detection each take 0.37 s, against 13 s for the invariant-feature scheme, 91 s for ContextLS and 176–190 sequential decoding steps for full rewriting. At equal latency no post-hoc baseline reaches our detection rate; at equal token-forwards TextSeal-Gumbel reaches 0.98–0.99 against our 0.94 by editing 79% of the tokens against 8%, and per changed token ReDiffMark carries z^2/edit ≈ 1.4 against 0.15–0.49 for full rewriting (Appendix D.3).

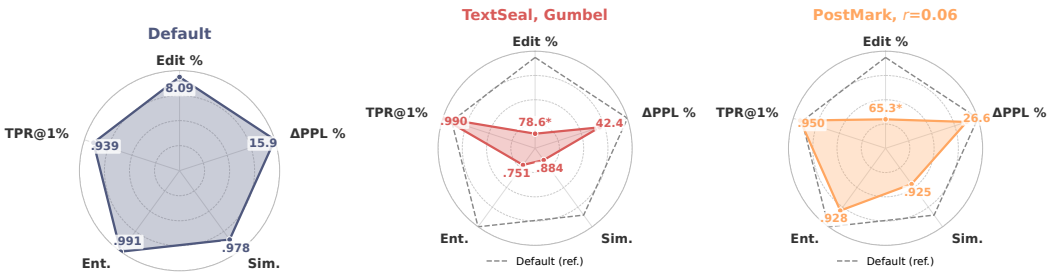


Figure 4: **Fidelity against detection.** One radar per method on 200-token C4 passages; outward is better and the dashed outline is the default point; all methods in Figure 11, numbers in Table 12.

5.4 Text Fidelity & Quality

The perplexity cost is that of resampling, not of the watermark (Figure 4). At identical positions the keyed refill costs 17.5% against 17.9% for plain resampling (Appendix B.4). Full rewriting changes 78% of the tokens and raises perplexity by 42–91% under one key, PostMark lengthens the passage by 43%, and the invariant-feature scheme changes about 2% of the tokens at five times our perplexity per edited token (Appendix D.4).

5.5 Robustness

Length-preserving edits degrade the watermark gracefully; length-changing edits defeat position keys but not left-context keys; re-translation remains the hard case (Figure 5). Each edit attack acts on a fraction $a \in \{0, 10, 20, 30, 40\}\%$ of the tokens, by deletion, uniform substitution, insertion, or washing, which re-diffuses a fraction a of the eligible positions with an unwatermarked LLaDA; panels (a)–(d) give the full ROC at every a, (e) the effect of back-translation at each ρ, and (f) the attacker's cost as the similarity of the attacked text to the watermarked one. Ten percent random substitution leaves TPR@1% at 0.59–0.66 while costing the attacker a similarity of 0.83; washing with an unwatermarked dLLM leaves 0.62–0.72 at 30% of the eligible positions and removes the watermark only at 80%. Deletion and insertion drive position keys to $z \approx 0$ at any strength, whereas left-context keys retain 0.68 at 10% and 0.48 at 20% deletion; back-translation leaves position keys at 0.04–0.05 and left-context keys at 0.18–0.36 (Appendix D.5).

5.6 Payloads, Provenance and Detection without the Model

The same key material carries a payload, attributes sources and admits a model-free detector. An 8-bit message is recovered with accuracy 0.21–0.82 depending only on the code (Appendix D.7); 400-token two-source documents are attributed with precision ≥ 0.96 and token-level segmentation accuracy up to 0.81 (Appendix D.8); a detector holding only the key and the tokenizer reaches TPR@1% $= 0.688$ at 0.03 s per passage on a CPU (Appendix D.7).

5.7 Ablation

Every component of §4 earns its place (Figure 6; Appendix D.6). At identical positions a green-list rule halves TPR@1% (0.94 to 0.28–0.44); masking all selected positions at once quadruples the perplexity cost; a RoBERTa-large infiller leaves detection unchanged but costs $\approx 1.6\times$ the perplexity; skipping the replay costs 0.25 of TPR@1%, and surprisal weighting adds 0.7–1.1 to z.

6 Conclusion and Limitations

We studied the problem of watermarking finished texts, where the goal is to locally and swiftly insert imperceptible watermarks at low cost. By shifting the paradigm to dLLM, we transformed it into a simple yet effective re-diffusion problem and introduced ReDiffMark: two checkerboard forward passes refill sparse positions with distortion-free continuous signals that tackle context entropy compression; then a replay detector re-derives the refill distributions only from the received text to address the asymmetry between the embedder and the receiver. We conducted comprehensive experiments, and our evaluation showed that ReDiffMark outperforms the baselines, achieving

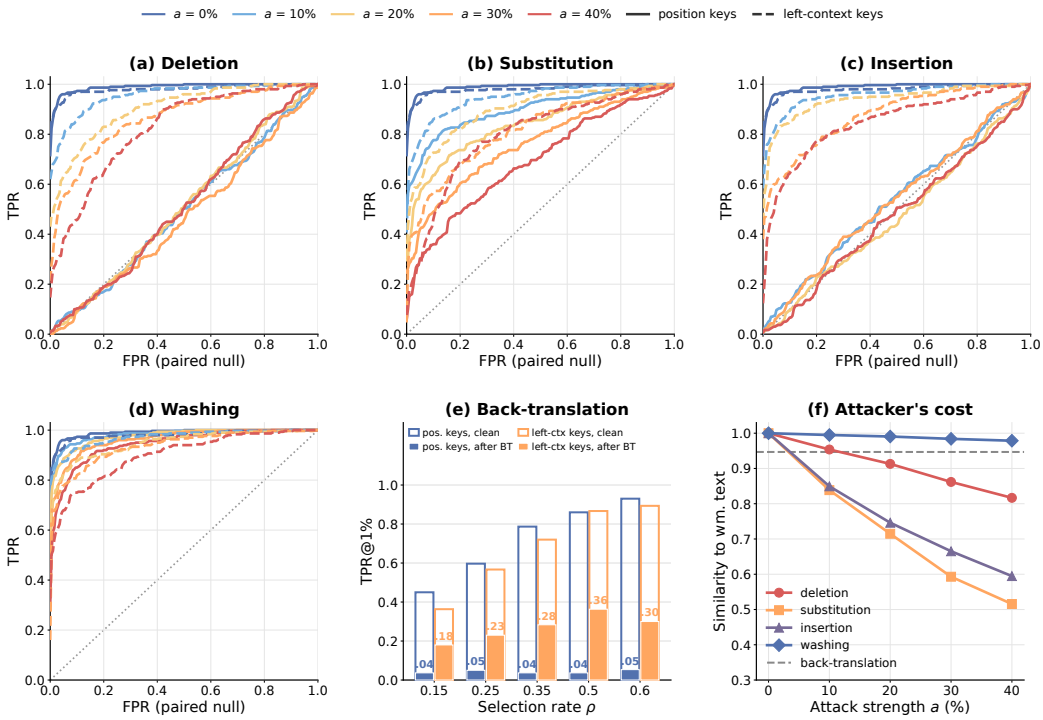


Figure 5: **Robustness to editing and re-translation.** Default point, 300 C4 passages, paired null; solid lines are position keys and dashed lines left-context keys, colours the attack strength a; the dotted diagonal is chance, and the same ROCs on a log FPR axis are in Figure 12.

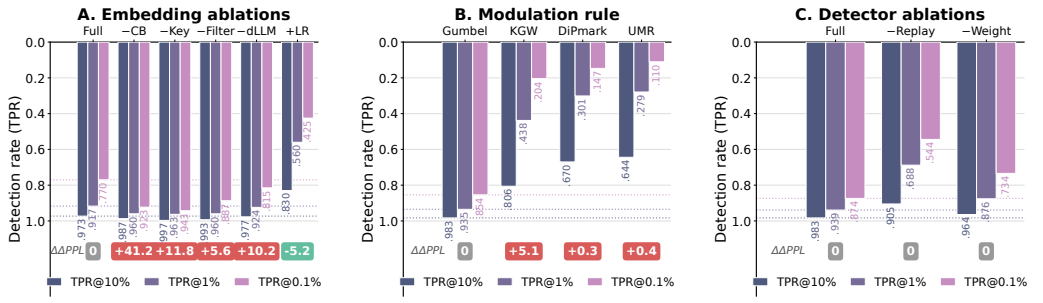


Figure 6: **Ablations.** TPR at three FPRs per variant and, as badges, the extra perplexity over the full method in points; −CB removes the checkerboard, −Key selects by entropy instead of by key, −Filter lets digits and capitalized words be edited, −dLLM uses RoBERTa-large, +LR adds the likelihood-ratio gate, −Replay scores without the model, −Weight drops surprisal weighting (Table 17).

excellent overall performance. This establishes ReDiffMark as a practical tool that can be applied to protect the rights of data owners without interfering with data creation or disrupting existing data. We hope our work provides a valuable perspective for future works on post-hoc watermarking, including combining ReDiffMark with other dLLM watermarks.

ReDiffMark is for natural-language prose stored verbatim or lightly edited, not a defense against free rewriting: a strong LLM paraphrase can remove almost the entire watermark. Low-entropy text, such as code or dense numerical data, limits the widespread use of ReDiffMark. Although the fidelity and requirements for post-hoc watermark protection are relatively weak in scenarios involving open-source and ambiguous code, and in dense data storage formats that are not often text-based, an effective solution for low-entropy text remains a meaningful future challenge.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, provide ingredients for proving them, and implement methods. We also used them to polish the paper for readability. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including any text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Watermarks for existing text can be used to attribute and track documents. The scheme presented here requires the secret key both to embed and to detect, so it cannot be used to identify text as machine-generated without the cooperation of the key holder, and it does not detect any property of the original author. Because the edits are sparse substitutions among near-synonyms chosen by a language model, watermarked text may occasionally differ in nuance from the original; applications that require exact wording (legal or medical text, quotations) should protect such spans with the rule filter.

REPRODUCIBILITY STATEMENT

The complete algorithm is given in Algorithms 1 and 2, hyper-parameters in Section 5.1 and Appendix A.1, and the derivation of the null distribution in Appendix B.1. All experiments use public data (C4 `realnewslike`, HotpotQA, GSM8K, MBPP) and open-weight models (LLaDA-8B, Dream-v0-Instruct-7B, Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct), with GPT-4o-mini called through its API for one text source and for the paraphrase and translation attacks; the data splits, generation settings and every prompt are given in Appendix A.2 and Appendix A.4. The implementation of the watermarker, detector, baselines, attacks, and evaluation scripts, together with the per-passage result files behind every table, will be released with the camera-ready version.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 17981–17993, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Dara Bahri and John Frederick Wieting. A watermark for black-box language models. *Trans. Mach. Learn. Res.*, 2026, 2026. URL https://openreview.net/forum?id=6gcHcgGmLo.

Yapei Chang, Kalpesh Krishna, Amir Houmansadr, John Wieting, and Mohit Iyyer. Postmark: A robust blackbox watermark for large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 8969–8987. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.506. URL https://doi.org/10.18653/v1/2024.emnlp-main.506.

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.

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.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. In *IEEE International Workshop on Information Forensics and Security, WIFS 2023, Nürnberg, Germany, December 4-7, 2023*, pp. 1–6. IEEE, 2023. doi: 10.1109/WIFS58808.2023.10374576. URL https://doi.org/10.1109/WIFS58808.2023.10374576.

Pierre Fernandez, Tom Sander, Hady Elsahar, Hongyan Chang, Tomáš Souček, Valeriu Lacatusu, Tuan A. Tran, Sylvestre-Alvise Rebuffi, and Alexandre Mourachko. How good is post-hoc watermarking with language model rephrasing? In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ZR1GOaadgP.

Thibaud Gloaguen, Robin Staab, Nikola Jovanovic, and Martin T. Vechev. Watermarking diffusion language models. *CoRR*, abs/2509.24368, 2025. doi: 10.48550/ARXIV.2509.24368. URL https://doi.org/10.48550/arXiv.2509.24368.

Sidong Guo, Tyler Kann, Teodora Baluta, and Matthieu R. Bloch. Covert multi-bit LLM watermarking: An information theory and coding approach. *CoRR*, abs/2605.16709, 2026. doi: 10.48550/ARXIV.2605.16709. URL https://doi.org/10.48550/arXiv.2605.16709.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. Protecting intellectual property of language generation apis with lexical watermark. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 10758–10766. AAAI Press, 2022a. doi: 10.1609/AAAI.V36I10.21321. URL https://doi.org/10.1609/aaai.v36i10.21321.

Xuanli He, Qiongkai Xu, Yi Zeng, Lingjuan Lyu, Fangzhao Wu, Jiwei Li, and Ruoxi Jia. CATER: intellectual property protection on text generation apis via conditional watermarks. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022b. URL http://papers.nips.cc/paper_files/paper/2022/hash/2433fec2144ccf5fea1c9c5ebdbc3924-Abstract-Conference.html.

Pyo Min Hong and Albert No. dgMARK: Decoding-guided watermarking for diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=mJJaz04lQR.

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.

Nikola Jovanovic, Robin Staab, Maximilian Baader, and Martin T. Vechev. Ward: Provable RAG dataset inference via LLM 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=kVrwHLAb20.

Joeun Kim, HoEun Kim, and Young-Sik Kim. CORE-BREW: llr-based soft decoding for robust multi-bit LLM watermarking. *CoRR*, abs/2606.24163, 2026. doi: 10.48550/ARXIV.2606.24163. URL https://doi.org/10.48550/arXiv.2606.24163.

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.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks 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=DEJIDCmWOz.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/575c450013d0e99e4b0ecf82bd1afaa4-Abstract-Conference.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.

Gregory Kang Ruey Lau, Xinyuan Niu, Hieu Dao, Jiangwei Chen, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Waterfall: Scalable framework for robust text watermarking and provenance for llms. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 20432–20466. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.1138. URL https://doi.org/10.18653/v1/2024.emnlp-main.1138.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. Who wrote this code? watermarking for code generation. 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. 4890–4911. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.268. URL https://doi.org/10.18653/v1/2024.acl-long.268.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. 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. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip S. Yu. A survey of text watermarking in the era of large language models. *ACM Comput. Surv.*, 57(2):47:1–47:36, 2025. doi: 10.1145/3691626. URL https://doi.org/10.1145/3691626.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. 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. 32819–32848. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. An entropy-based text watermarking detection method. 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. 11724–11735. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.630. URL https://doi.org/10.18653/v1/2024.acl-long.630.

Travis J. E. Munyer and Xin Zhong. Deeptextmark: Deep learning based text watermarking for detection of large language model generated text. *CoRR*, abs/2305.05773, 2023. doi: 10.48550/ARXIV.2305.05773. URL https://doi.org/10.48550/arXiv.2305.05773.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

Gang-Hyun Park, Ju-Hyeong Lee, Hee-Youl Kwak, and Dae-Young Yun. Weavemark: Robust and scalable multi-bit llm watermarking via coded payload spreading, 2026. URL https://arxiv.org/abs/2609.02177.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. Provably robust multi-bit watermarking for ai-generated text. In Lujo Bauer and Giancarlo Pellegrino (eds.), *34th USENIX Security Symposium, USENIX Security 2025, Seattle, WA, USA, August 13-15, 2025*, pp. 201–220. USENIX Association, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/qu-watermarking.

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.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can ai-generated text be reliably detected? stress testing AI text detectors under various attacks, 2025. URL https://openreview.net/forum?id=OOgsAZdFOt.

Subham S. Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. 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/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Tom Sander, Hongyan Chang, Tomáš Souček, 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. *arXiv preprint arXiv:2605.12456*, 2026.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. 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/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Alexandr Goultiaev Tolstokorov, Kyriakos Mouratidis, Javad Dogani, and Nikolaos Laoutaris. Rent-a-rag: Embedding-space watermarks for auditing third-party rag, 2026. URL https://arxiv.org/abs/2609.03749.

Umut Topkara, Mercan Topkara, and Mikhail J. Atallah. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In Sviatoslav Voloshynovskiy, Jana Dittmann, and Jessica J. Fridrich (eds.), *Proceedings of the 8th workshop on Multimedia & Security, MM&Sec 2006, Geneva, Switzerland, September 26-27, 2006*, pp. 164–174. ACM, 2006. doi: 10.1145/1161366.1161397. URL https://doi.org/10.1145/1161366.1161397.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. Towards codable watermarking for injecting multi-bits information to llms. 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=JYu5Flqm9D.

Yidan Wang, Yubing Ren, Yanan Cao, and Li Guo. Worldcup sampling for multi-bit LLM watermarking. *CoRR*, abs/2602.01752, 2026. doi: 10.48550/ARXIV.2602.01752. URL https://doi.org/10.48550/arXiv.2602.01752.

Linyu Wu, Linhao Zhong, Wenjie Qu, Yuexin Li, Yue Liu, Shengfang Zhai, Chunhua Shen, and Jiaheng Zhang. Dmark: Order-agnostic watermarking for diffusion large language models. *CoRR*, abs/2510.02902, 2025. doi: 10.48550/ARXIV.2510.02902. URL https://doi.org/10.48550/arXiv.2510.02902.

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.

Ke Yang, Dongyang Liang, Jing Yu, Shuguang Yuan, and Chi Chen. You can have a second chance: Unbiased and multi-bit watermarking for diffusion language models with regret-based remasking. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28141–28160, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1297. URL https://aclanthology.org/2026.acl-long.1297/.

Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. Tracing text provenance via context-aware lexical substitution. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 11613–11621. AAAI Press, 2022. doi: 10.1609/AAAI.V36I10.21415. URL https://doi.org/10.1609/aaai.v36i10.21415.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1259. URL https://doi.org/10.18653/v1/d18-1259.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *CoRR*, abs/2508.15487, 2025. doi: 10.48550/ARXIV.2508.15487. URL https://doi.org/10.48550/arXiv.2508.15487.

KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. Robust multi-bit natural language watermarking through invariant features. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 2092–2115. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.117. URL https://doi.org/10.18653/v1/2023.acl-long.117.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for large language models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 4031–4055. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.224. URL https://doi.org/10.18653/v1/2024.naacl-long.224.

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for 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. 58851–58880. PMLR / OpenReview.net, 2024a. URL https://proceedings.mlr.press/v235/zhang24o.html.

Ruisi Zhang, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. REMARK-LLM: A robust and efficient watermarking framework for generative large language models. In Davide Balzarotti and Wenyuan Xu (eds.), *33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*. USENIX Association, 2024b. URL https://www.usenix.org/conference/usenixsecurity24/presentation/zhang-ruisi.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. 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=SsmT8aO45L.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 11425–11460. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.524. URL https://doi.org/10.18653/v1/2026.acl-long.524.

A Implementation Details

Algorithms, hyper-parameters and metric definitions, the implementation of every baseline, and every prompt used in the paper.

A.1 Algorithms and hyper-parameters

Algorithms. Algorithm 1 gives the embedder of Section 4.1 and Algorithm 2 the model-assisted detector of Section 4.5 in full.

Algorithm 1 ReDiffMark embedding (zero-bit; $L = 1$)

Require: text $\boldsymbol{x} \in \mathcal{V}^n$, master key sk, dLLM $\mathcal{M}$, rate ρ, gate $(h_{\min}, k, p_{\text{gate}})$, nucleus p_{s}

$\mathsf{k}_{\text{sel}}, \mathsf{k}_{\text{g}} \leftarrow \text{HMAC}(\mathsf{sk}, \texttt{"select"}),\ \text{HMAC}(\mathsf{sk}, \texttt{"global"})$
$\mathcal{S} \leftarrow \{\, i \in \mathcal{R}(\boldsymbol{x}) :\ \text{PRF}(\mathsf{k}_{\text{sel}}, i) < \rho \,\}$ ▷ text-only rules + keyed selection
for parity $\pi \in \{0, 1\}$ **do** ▷ checkerboard: two forward passes in total
  $\mathbb{S}_\pi \leftarrow \{i \in \mathcal{S} : i \bmod 2 = \pi\}$; obtain $\{p_i\}_{i \in \mathbb{S}_\pi}$ from *one* forward pass of $\mathcal{M}$ with $\mathbb{S}_\pi$ masked
  for $i \in \mathbb{S}_\pi$ **in parallel do**
    if $\text{H}(p_i) \geq h_{\min}$ **and** $x_i \in \text{TopK}(p_i, k) \cup \text{Nucleus}(p_i, p_{\text{gate}})$ **then** ▷ model gate, Equation 3
      $q_i \leftarrow$ top-p truncation of p_i at p_{s}; $u_{i,\cdot} \leftarrow \text{PRF}(\mathsf{k}_{\text{g}}, \texttt{"gumbel"}, i)$
      $x_i \leftarrow \arg\max_v \log u_{i,v} / q_{i,v}$ ▷ distortion-free, Equation 4
    end if
  end for
end for
return $\boldsymbol{x}$ ▷ all positions outside $\mathcal{S}$ are verbatim

Algorithm 2 ReDiffMark model-assisted detection by replay

Require: received text $\boldsymbol{y} \in \mathcal{V}^n$, sk, $\mathcal{M}$, ρ, $h_{\min}$, $w_{\max}$

recompute $\mathsf{k}_{\text{sel}}, \mathsf{k}_{\text{g}}$ and $\mathcal{S} \leftarrow \{i \in \mathcal{R}(\boldsymbol{y}) : \text{PRF}(\mathsf{k}_{\text{sel}}, i) < \rho\}$
for parity $\pi \in \{0, 1\}$ **do** ▷ replay the embedding schedule on $\boldsymbol{y}$
  obtain $\{\hat{p}_i\}_{i \in \mathbb{S}_\pi}$ from one forward pass of $\mathcal{M}$ with $\mathbb{S}_\pi = \{i \in \mathcal{S} : i \bmod 2 = \pi\}$ masked
  for $i \in \mathbb{S}_\pi$ with $\text{H}(\hat{p}_i) \geq h_{\min}$ **do** ▷ replayed entropy gate
    $u_i \leftarrow \text{PRF}(\mathsf{k}_{\text{g}}, \texttt{"gumbel"}, i)_{y_i}$; $w_i \leftarrow \text{clip}(-\log \hat{p}_i(y_i), 0, w_{\max})$
  end for
end for
$S \leftarrow \sum_i w_i \,(-\log(1 - u_i))$; $\kappa \leftarrow (\sum_i w_i)^2 / \sum_i w_i^2$; $\theta \leftarrow \sum_i w_i^2 / \sum_i w_i$
return $z = \Phi^{-1}\big(1 - \Pr[\text{Gamma}(\kappa, \theta) \geq S]\big)$ ▷ Proposition 1

Key derivation. A master secret sk is never used directly. Every purpose-specific key is $\text{HMAC}_{\text{BLAKE2b}}(\mathsf{sk},\ \texttt{"derive"} \,\|\, \text{purpose} \,\|\, \text{identifier})$: k_{sel} (purpose `select`), k_{g} (`global`), and k_s (`source`, identifier s). $\text{PRF}(\mathsf{k}, \cdot)$ maps the first 8 bytes of $\text{HMAC}(\mathsf{k}, \cdot)$ to $[0, 1)$; the Gumbel uniform vector of a position is drawn from a PRNG seeded with $\text{PRF}(\mathsf{k}_{\text{g}}, \texttt{"gumbel"}, i)$ (and, for multi-bit, with the position's symbol appended; Appendix D.7), so that the embedder and the detector obtain identical vectors.

Rule filter. A position is eligible iff it is not the first token; its surface form contains no digit; it is not punctuation-only; it starts a word (Ġ/▁/whitespace-prefixed for BPE/SentencePiece tokenizers) and is not followed by a sub-word continuation (so that no word is ever broken); it is not capitalised unless the previous token ends a sentence; and it does not fall inside a URL, a code span (`` `...` ``), an inline formula (`$...$`), or a user-supplied protected span. The filter is deterministic in the token strings and is recomputed by the detector.

Schedule. For $L = 1$ the two checkerboard passes mask the selected positions of even and odd index respectively; for general L the blocks $[kL, (k+1)L)$ are assigned by the parity of k. In the

sequential variant the positions of every fragment are decoded left to right, all fragments of a pass advancing together, so a pass costs L forward passes. A hard budget of $\lfloor 0.25\,n \rfloor$ changed tokens is enforced; it was never reached in our experiments. Texts of equal token length are processed in lock-step so that every forward pass is a rectangular batch.

Instruct wrapping. For LLaDA-8B-Instruct the text is placed in the assistant turn of the tokenizer's chat template, `[user:` *`Reproduce the following passage exactly.`*`][assistant:` $\boldsymbol{x}$`][eot]`; the prefix and suffix are added inside the backend and sliced off, so the watermarker and detector only ever see the text's own token indices. Without the wrapper the Instruct model tends to emit end-of-turn tokens at masked positions.

A.2 METRICS

Fidelity. All fidelity metrics compare a watermarked passage with its original. Edit % is the fraction of token positions whose token changed; for methods that rewrite the passage, whose output no longer aligns position by position, it is the token edit distance in percent of the original length and is marked *. ΔPPL % is the mean paired relative increase in perplexity under an independent Llama-3.1-8B (base). Sim. is the cosine similarity of all-MiniLM-L6-v2 sentence embeddings (Reimers & Gurevych, 2019). Ent. is the fraction of capitalised words and numbers of the original that survive verbatim.

Detectability. z is the mean detection statistic of Equation 5; TPR@10%, TPR@1% and TPR@0.1% are true-positive rates at thresholds read off the empirical null of unwatermarked passages. The null FPR at $z \geq 4$ is reported alongside as a check on Proposition 1; the 0.1% tail rests on 1,000 null passages per key in the five-key grid (5,000 scored nulls in total) and is not reported where the null holds fewer than 1,000 passages, except for discrete-score baselines explicitly marked with the available null size. For the baselines whose score is discrete (bit matches, presence fractions) the threshold is the smallest score whose null exceedance is at most the target FPR.

Robustness. After an attack, TPR@1% is read against a *paired* null in which the same attack is applied to the unwatermarked originals, so that the threshold moves with the attack. Sub. is 10% random token substitution and BT is back-translation (en→de→en); the attacker's cost is the sentence similarity between the attacked and the unattacked watermarked passage.

Cost. Forward passes count sequences pushed through a model, steps count sequential model calls, and token-forwards count single-token forward equivalents, a FLOPs proxy in which a KV-cached decoding step counts one and a full-sequence forward its length; s/p is wall-clock seconds per passage at batch 1 on one shared A800-80GB. The composite Σ of Table 1 is the unweighted mean of one score per column group, scaled to 100 (Table 5).

Keys and deviations. Where a number carries $\pm$ it is the mean over independent master keys and $\pm$ the across-key standard deviation (five keys for the position-key rows of Table 1, three otherwise); numbers without $\pm$ are single-key runs (key 0).

Evaluation data. Generality uses 300 passages each of Wikipedia paragraphs (HotpotQA supporting documents, Yang et al., 2018), of C4-prefix continuations written by Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct and GPT-4o-mini (Appendix A.4.1), of C4 at 100, 300 and 500 tokens, of GSM8K solutions and of MBPP programs, and 400-token documents that alternate two Wikipedia sources for provenance. Perplexity is scored by Llama-3.1-8B base, and the paraphrase and translation attacks use GPT-4o-mini (Appendix A.4.3).

Baselines. The green-list baselines replace only the modulation of Equation 4: UMR reweights p by $(1+\delta)$ on the green set and by $(1-\delta\tau/(1-\tau))$ off it, with $\tau = p(\text{green})$ and δ clipped so that the result stays a distribution; DiPmark reweights the CDF along a key permutation of the vocabulary with parameter α; KGW adds δ to the green logits. All use $\gamma = 0.5$ and position-indexed green sets, and are detected by the usual green-count $z = (G - \gamma T)/\sqrt{T\gamma(1-\gamma)}$ over the same replayed, entropy-gated positions.

Table 2: Default hyper-parameters.

Component	Parameter	Default
Selection	rate ρ (default point; swept in Table 7)	0.5
	fragment length L	1
Model gate	entropy threshold $h_{\min}$ (nats)	0.5
	candidate set: top-k / nucleus p_{gate}	20 / 0.95
	edit cap (fraction of tokens)	0.25
Modulation	sampling nucleus p_{s}	0.9 (1.0 = none)
	temperature	1.0
Detection	replayed entropy gate $h_{\min}$	0.5
	weight clip $w_{\max}$ / offset	3 / 0
	candidate-set gate at detection	off
Provenance	emission gain δ / self-transition σ / null prior	0.8 / 0.97 / 0.5
Baselines	UMR δ / DiPmark α / KGW δ / γ	1 / 0.45 / 2 / 0.5

A.3 POST-HOC BASELINES

PostMark. We run the authors' code for word selection (nomic-embed-text-v1 with the secret embedding table, $k = \lfloor r\,|\text{words}| \rfloor$ words at insertion ratio r), for the insertion prompt and the iterative insertion loop (sub-lists of ten words, up to three attempts until half of a sub-list is present), and for the presence-score detector (Paragram cosine ≥ 0.7 on words re-selected from the received text). The two resources that the authors distribute through a Google Drive folder are no longer available and are rebuilt: the Paragram vectors come from the Paragram-SL999 release (1,703,187 of the 1,703,444 words of PostMark's vocabulary are covered, the rest are zero vectors as in PostMark), and the secret table samples embeddings of 20,000 sentence-bounded C4 training passages instead of RedPajama passages. The key is the seed of the table sampling. The loop is batched across passages, and a retry stops when greedy decoding repeats its output. Llama-3.1-8B-Instruct (greedy) inserts the words in every configuration of the paper; the authors' GPT-4o inserter was run once at $r = 0.12$ for comparison ($\text{TPR@1\%} = 1.000$, $\Delta\text{PPL} = 49.9\%$, passage 74% longer).

PostMark's insertion budget. PostMark's insertion ratio sets how many watermark words are drawn ($r \cdot \#\text{words}$) and then written into the passage by regenerating it around them, so r controls how much of the original survives. At the authors' default $r = 0.12$ with the same local inserter the passage grows by 88%, nothing is preserved in place (token edit distance 113%), perplexity rises by 28.5% and embedding costs 34 s per passage, which buys $\text{TPR@1\%} = 0.983$. That operating point is close to unconstrained rewriting, so Table 1 reports the tighter $r = 0.06$ instead: the passage grows by 43%, the edit distance falls to 65.3%, ΔPPL to 26.6% and the wall-clock to 5.9 s, at $\text{TPR@1\%} = 0.950$ and $\text{TPR@0.1\%} = 0.877$ under one key (8.5 inserted words per passage, 1.07 inserter calls; three-seed means in Table 1). Halving the budget therefore costs 0.03 of TPR@1% while cutting the latency six-fold and returning a third of the passage to verbatim, which is the trade a reader should see: the default setting's detection advantage comes from regeneration rather than from the watermark.

ContextLS. We run the authors' implementation (Yang et al., 2022) unchanged for everything that defines the method: the BERT fill-mask candidate generator, the synchronicity test that keeps only positions whose candidate set is stable under substitution, and the substitutability test that keeps the extractor in step; the walk over positions is theirs. The scheme defines no presence test, so we make it zero-bit exactly as we do for Yoo et al. (2023): the bit carried by the j-th embeddable position is $\text{PRF}(\mathsf{sk}, j)$ rather than a random bit, the detector re-runs the same walk on the received passage and recovers one bit per embeddable position, and the statistic is $z = (\text{matches} - B/2)/\sqrt{B/4}$ over the B recovered bits, thresholded against the same empirical null of 1,000 unwatermarked passages under the strict rule, since the score is discrete.

Invariant-feature substitution. We run the authors' code with the settings of their released script (keyword ratio 0.05, top-4 BERT-base-cased candidates, grammar-component masks ordered by dependency label, no robust-infill checkpoint), including the validity check that re-parses every candidate combination. The scheme extracts a multi-bit message but defines no presence test. We

therefore embed in sentence j the first $b_j = \lfloor \log_2 |V_j| \rfloor$ bits of $\mathrm{PRF}(\mathsf{k}, j)$, where V_j is the set of valid watermarked variants (the original uses $\lceil \cdot \rceil$ and random bits), and score a received passage by $z = (m - B/2)/\sqrt{B/4}$ over its B extracted bits with m matches. Passages carry 10.3 bits on average (0.108 bits per word) and are extracted with 99.0% bit accuracy, against 47.5% on unwatermarked passages.

RoBERTa-large infiller. The masked-LM backend replaces only the infill logits; RoBERTa's byte-level tokenizer is lossless, so text outside the edits is still reproduced verbatim (99% of watermarked passages re-tokenise exactly).

A.4 Prompts

Every LLM in the pipeline other than the dLLM is driven by one fixed prompt, reproduced here verbatim from the code (line breaks added for the page; text in braces is filled per passage). Each prompt is sent as a single user turn, through the tokenizer's chat template for local models.

A.4.1 Text-source generation

Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct and GPT-4o-mini continue the first 25 words of a C4 passage taken from the null pool, so that no source overlaps the evaluation set; the prefix is kept, and the text is cut to 200 LLaDA tokens. The local models sample at temperature 0.7, top-p 0.9, at most 360 new tokens; GPT-4o-mini at temperature 0.7, at most 450 tokens.

```
Continue writing the following news article in the same style. Write at
least 300 words of plain prose, no headings, no lists, no preamble.

{first 25 words of the C4 passage}
```

A.4.2 Baselines

TextSeal rephrasing (Llama-3.1-8B-Instruct under KGW or Gumbel sampling). The prompt of our implementation of the rephrasing baseline; sampling at temperature 1 without truncation, the watermark acting on the next-token distribution, at most 320 new tokens.

```
Rewrite the following passage in your own words. Keep every fact, name,
number and quotation, keep roughly the same length, and output only the
rewritten passage.

{passage}
```

PostMark word insertion (Llama-3.1-8B-Instruct). Taken from the authors' code (Chang et al., 2024); the word list is passed in sub-lists of ten words per call, a call is repeated up to three times while fewer than half of its words appear in the output, decoding is greedy with at most 700 new tokens.

```
Given below are a piece of text and a word list. Rewrite the text to
incorporate all words from the provided word list. The rewritten text
must be coherent and factual. Distribute the words from the list evenly
throughout the text, rather than clustering them in a single section.
When rewriting the text, try your best to minimize text length increase.
Only return the rewritten text in your response, do not say anything
else.

Text: {passage}

Word list: {words}

Rewritten text:
```

A.4.3 Attacks

All attacks use GPT-4o-mini at temperature 0.7 with at most 600 output tokens; outputs are cached by the hash of the input, so the paired null and every detector see the same attacked text.

Paraphrase (one hop; the three-hop chain feeds the output back in).

```
Paraphrase the following passage sentence by sentence. Keep the meaning,
all names, numbers and facts; change the wording and sentence structure
as much as possible. Output only the paraphrased passage.

{passage}
```

Back-translation (en→de→en).

```
Translate the following English passage into German. Output only the
translation.

{passage}

Translate the following German passage into English. Output only the
translation.

{German passage}
```

A.4.4 LLaDA-8B-Instruct wrapper

For LLaDA-8B-Instruct the passage is placed in the assistant turn of the tokenizer's chat template (Appendix A.1); the template's prefix and suffix are added inside the backend and sliced off again, so the watermarker and the detector index only the passage's own tokens.

```
[user]       Reproduce the following passage exactly.
[assistant]  {passage}
[end of turn]
```

B ReDiffMark: Statistical Analysis

Why the detector is calibrated and how much signal a post-hoc edit can carry: the null distribution, the entropy budget of post-hoc re-diffusion, the choice of detection statistic, and the perplexity decomposition of Section 5.4.

B.1 Null distribution and calibration

Proof of Proposition 1. Under H_0 the received text $\boldsymbol{y}$ was produced without k_g, so the PRF outputs $u_{i,\cdot} = \mathrm{PRF}(\mathsf{k}_\mathrm{g}, \texttt{"gumbel"}, i)$ are (computationally indistinguishable from) i.i.d. uniform vectors independent of $\boldsymbol{y}$; in particular $u_i = u_{i,y_i} \sim \mathrm{Unif}(0,1)$ independently across the scored positions, whatever the dependence structure of $\boldsymbol{y}$. Hence $E_i = -\log(1-u_i) \sim \mathrm{Exp}(1)$ i.i.d. The weights $w_i = \mathrm{clip}(-\log \hat{p}_i(y_i), 0, w_{\max})$ and the scored set are functions of $\boldsymbol{y}$ and of k_sel only, hence independent of the E_i. Conditionally on $\boldsymbol{y}$, $S = \sum_i w_i E_i$ is a linear combination of independent exponentials with $\mathbb{E}[S] = \sum_i w_i$ and $\mathrm{Var}[S] = \sum_i w_i^2$; a $\mathrm{Gamma}(\kappa, \theta)$ with $\kappa\theta = \sum_i w_i$ and $\kappa\theta^2 = \sum_i w_i^2$ matches these moments, which gives $\kappa = (\sum_i w_i)^2 / \sum_i w_i^2$ and $\theta = \sum_i w_i^2 / \sum_i w_i$, and is exact when all w_i are equal (then $S/w \sim \mathrm{Gamma}(T, 1)$). The exact conditional law of a general linear combination is hypoexponential (generalised Erlang) with rates $1/w_i$, whose tail decays as $e^{-s/w_{\max}}$, whereas the moment-matched Gamma decays as $e^{-s/\theta}$ with $\theta = \sum_i w_i^2 / \sum_i w_i \le w_{\max}$: the approximation is exact for equal weights, matches the first two moments otherwise, and is lighter-tailed than the exact law, so the z it reports is slightly optimistic far out in the tail; the clip $w_{\max} = 3$ bounds the gap. In the operating regime ($T \approx 25$ scored positions, $\kappa \approx 15$–20, thresholds two to three standard deviations above the mean) the gap is small: on the 1,000 unwatermarked C4 passages

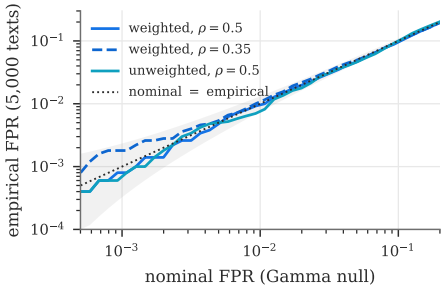


Figure 7: Empirical against nominal calibration of the detector.

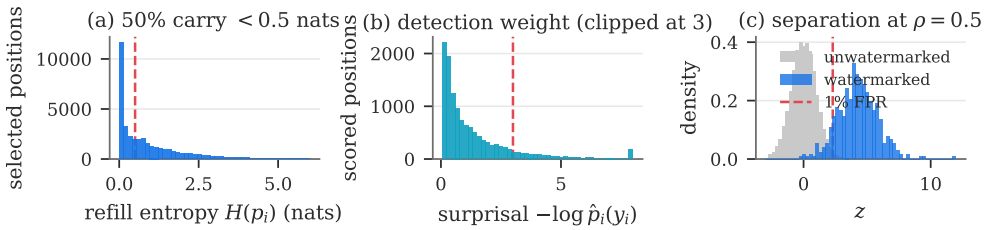


Figure 8: *(Left)* Refill entropy at key-selected positions of unwatermarked passages ($\rho = 0.35$). *(Middle)* Surprisal of the received token under the replayed distribution. *(Right)* Statistic separates passages at 1% FPR on 91% of them ($\rho = 0.5$, key 0).

of key 0, the hypoexponential tail (Monte Carlo, 10^6 draws per passage) at the Gamma threshold of nominal FPR 1% is 1.16% on average (95th percentile over passages 1.28%), and at nominal 0.1% it is 0.15% (0.19%); the exact 0.1% quantile exceeds the Gamma one by 2.7% on average (`scripts/weight_theory_check.py`). The empirical check of Figure 7 agrees, and every TPR in the paper is read against empirical null thresholds, so the Gamma enters only the reported z values.

Empirical check. Figure 7 checks the approximation on $5{,}000$ unwatermarked C4 passages (five keys): the empirical FPR against the nominal FPR implied by the moment-matched Gamma null, for the surprisal-weighted statistic at $\rho = 0.5$ and 0.35 and for the unweighted statistic whose null is exactly $\mathrm{Gamma}(T, 1)$; the shaded band is the 95% binomial interval for $5{,}000$ samples, and every curve stays inside it down to 10^{-3}. The empirical 1% thresholds of the five keys at $\rho = 0.5$ lie in $[2.16, 2.43]$ against the nominal 2.33, so a threshold set on one key transfers to another with a change in FPR of a fraction of a percent.

Expected score under H_1. If y was drawn by Equation 2 with probability $p = q_y$, then $u_y \overset{d}{=} U^p$ with $U \sim \mathrm{Unif}(0, 1)$ (the maximum $\max_v u_v^{1/p_v}$ is uniform and independent of the arg max), so

$$\mathbb{E}\big[-\log(1 - u_y)\big] = \sum_{k \geq 1} \frac{\mathbb{E}[U^{pk}]}{k} = \sum_{k \geq 1} \frac{1}{k(pk + 1)} = \mathcal{H}_{1/p} \;\approx\; \log \frac{1}{p} + \gamma_{\mathrm{E}}, \tag{6}$$

with $\gamma_{\mathrm{E}} \approx 0.577$ the Euler–Mascheroni constant. The mean shift of a scored position is therefore exactly $\delta(p_y) = \mathcal{H}_{1/p_y} - 1$, while its null variance is 1; among linear statistics $S = \sum_i w_i E_i$ the one that maximises the standardised shift $\sum_i w_i \delta_i / \sqrt{\sum_i w_i^2}$ has $w_i \propto \delta_i$, so the locally optimal weights are the exact shifts. We use their leading term $-\log p$, which differs from $\delta(p)$ by $\gamma_{\mathrm{E}} - 1 \approx -0.42$ for small p and vanishes with it at $p = 1$ ($\delta = 0.50, 1.28, 1.93, 2.60$ against $-\log p = 0.69, 1.61, 2.30, 3.00$ at $p = 0.5, 0.2, 0.1, 0.05$); the efficiency of a linear statistic with mis-specified weights is the squared cosine between the two weight vectors, which on the replayed $\hat{p}$ of the scored positions of the 500 watermarked passages of key 0 is 0.997 on average and never below 0.994 (`scripts/weight_theory_check.py`), so the leading term costs nothing measurable, and the replayed $\hat{p}_i$ stands in for the unknown q_i. Clipping at $w_{\max} = 3$ guards against positions where the replay differs from the embedding-time distribution (e.g. because a neighbouring position was itself edited).

B.2 SIGNAL BUDGET OF POST-HOC RE-DIFFUSION

Table 3: Signal budget (medians over texts; $\rho = 0.35$; five keys; LLaDA-8B-Base; unwatermarked refill). T = rule-eligible positions; selected = keyed selection; T_e = positions passing the entropy gate at $h_{\min}$; changed = positions where the refill differs from the original. C4 passages have 200 tokens; HotpotQA supporting sentences (60–80 tokens) and MBPP programs are used at their native length.

Data	$h_{\min}$	T	Selected	T_e	T_e/T	Changed
C4 ($n = 500$)	0.5	116	41	20	0.17	11
C4	1.0	116	41	13	0.11	9
C4	1.5	116	41	8	0.07	6
C4	2.0	116	41	5	0.04	4
HotpotQA sentences ($n = 300$)	0.5	29.5	11	4	0.14	2
HotpotQA sentences	1.0	29.5	11	2	0.07	1
MBPP code ($n = 300$)	0.5	8	3	0	0	0

Table 3 gives the counts behind the model gate of Section 4.2 and the replay of Section 4.5: raising the gate from 0.5 to 1.0 nats halves the scored positions again, and short supporting sentences and code expose too few positions for single-passage detection, which is why the main experiments use 200-token passages (Section 5.2 for code and arithmetic).

B.3 DETECTION STATISTICS

Detection range and weighting. The three detection ranges of Section 4.5 were measured on one embedding (LLaDA-8B-Base, no truncation, $L = 1$, $\rho = 0.35$, $n = 100$, five keys): scoring every rule-eligible position ($T = 109$) gives $z = 2.18$ and a detection rate of 0.084 at $z \geq 4$; scoring only the recomputed selection $\mathcal{S}$ ($T = 39.6$) gives $z = 3.27$ and 0.314; adding the replayed entropy gate ($T = 20.3$) gives $z = 4.04$ and 0.502, so the noise of unscored positions costs 5–9× the signal. Surprisal weighting on the same embedding ($n = 200$) raises z from 4.38 to 5.32 and TPR@0.1% from 0.76 to 0.905 with 0/1,000 null false positives at $z \geq 4$ and an empirical 1% threshold of 2.32. Adding a candidate-set gate at detection (requiring the received token to lie in the top-k/nucleus of $\hat{p}_i$) was slightly harmful and is disabled.

Exact likelihood ratio. The exact per-position Neyman–Pearson log-likelihood ratio for $u_y \sim \mathrm{Beta}(1/p_y, 1)$ versus uniform, $\mathrm{llr}_i = -\log p_y + \frac{1-p_y}{p_y} \log u_y$, has weights $(1 - p_y)/p_y$ that are dominated by a few positions with tiny $\hat{p}_i(y_i)$; even with clipping it reached only $z = 2.63$ where the surprisal-weighted statistic reached 5.3.

B.4 PERPLEXITY DECOMPOSITION

Table 4: Decomposition of the perplexity cost ($n = 200$, $\rho = 0.35$, $L = 1$; the same positions are refilled with different rules; z is the surprisal-weighted statistic; 95% bootstrap CIs in brackets). The watermarked Gumbel refill costs the same perplexity as unwatermarked multinomial resampling from the same distribution, on both dLLMs: modulation is free, the cost is resampling itself.

			LLaDA-8B-Base		LLaDA-8B-Instruct	
Refill rule	What it measures	WM	ΔPPL %	z	ΔPPL %	z
argmax	model mismatch floor	no	1.6 [-0.3, 3.9]	≈ 0	4.0 [1.5, 7.3]	≈ 0
multinomial	resampling floor	no	17.9 [15.2, 21.1]	≈ 0	27.4 [23.2, 32.7]	≈ 0
multinomial, top-p 0.9	resampling floor, truncated	no	11.6 [9.1, 14.6]	≈ 0	16.1 [13.0, 19.6]	≈ 0
Gumbel	ReDiffMark, no truncation	yes	17.5 [14.9, 20.7]	5.32	28.3 [24.2, 32.8]	5.13
Gumbel, top-p 0.9	ReDiffMark, default rule	yes	11.1 [8.7, 13.9]	4.03	16.5 [13.0, 20.5]	4.35

C EXPERIMENTAL DETAILS

Protocols that the main text only names: the attack setting and the two ways of setting a detection threshold, the composite score of Table 1, and the external-verifier variant.

C.1 Attacks and detection thresholds

We evaluate ReDiffMark in two key modes, position keys and left-context keys (Section 4.4), on 300 C4 passages at the default point under one key. Every attack is applied both to the watermarked passages and to their unwatermarked originals, and detection is read against that paired null, whose FPR at $z \geq 4$ is 0 for every attack.

Fixed threshold. The paired null answers whether an attack can be told apart from the same attack on unwatermarked text; a deployed detector instead fixes its threshold once, on clean unwatermarked text, and never sees the attack.

C.2 Composite score

Table 5: Composite score of Table 1. f_1, f_2, f_3 are the unweighted means of the fidelity columns (as $1-$Edit, $1-\Delta$PPL clipped to $[0,1]$, Sim., Ent.), of the two detectability columns and of the two robustness columns; $c = 1 - \log_{10}(s/0.2)/\log_{10} 300$ clipped to $[0,1]$; $\Sigma = 100 \cdot \mathrm{mean}(f_1, f_2, f_3, c)$. No weight is tuned and every row recomputes from the means in Table 1.

Method	f_1	f_2	f_3	c	Σ
ContextLS	.926	.480	.073	.000	37.0
Yoo et al.	.942	.851	.067	.263	53.1
PostMark, r=0.06	.732	.910	.726	.407	69.4
TextSeal, KGW	.461	.903	.617	.316	57.4
TextSeal, Gumbel	.458	.994	.894	.596	73.6
ReDiffMark, ρ=0.5	.932	.907	.342	.892	76.8
ReDiffMark, ρ=0.5, left-ctx	.934	.870	.519	.878	**80.0**

C.3 External verification

External verification. The implementation includes a reverse-speculative hook in which a large autoregressive model scores every proposed edit and rejects those whose log-likelihood ratio over a 16-token window falls below $-\varepsilon$, bounding the drift with respect to the verifier's distribution by roughly ε per accepted edit (Leviathan et al., 2023). Table 6 evaluates it on the 300 Llama-3.1-8B-Instruct continuations of Section 5.2, with Llama-3.1-8B (base) as the verifier and ΔPPL measured under the *generator's* own model: the verifier does what it promises, but it accepts only 33–56% of the proposed substitutions, and the rejected ones are precisely the surprising tokens that carry the Gumbel signal. This is the same phenomenon as the dLLM's own likelihood-ratio gate (Table 16): in a distortion-free scheme the signal and the perplexity cost are carried by the same tokens, so any filter that removes cost removes signal (Section 6).

Table 6: External verification (Llama-3.1-8B base verifier, window 16) on Llama-3.1-8B-Instruct continuations ($n = 300$, default point, one key; ΔPPL under the generator Llama-3.1-8B-Instruct; paired null).

Verifier	ε	Acceptance	Edit %	$\Delta\mathrm{PPL}_{\mathrm{gen}}$ %	z	TPR@10%	TPR@1%	s/passage
none	—	1.000	3.93	13.0	2.85	0.943	0.713	0.35
Llama-3.1-8B	0.5	0.328	1.28	0.3	0.65	0.343	0.083	0.85
Llama-3.1-8B	1.0	0.402	1.57	0.8	0.81	0.410	0.130	0.84
Llama-3.1-8B	2.0	0.562	2.20	2.2	1.29	0.573	0.227	0.84

D Additional Results

Full tables and figures behind each subsection of Section 5, in the order of the main text.

D.1 SELECTION-RATE GRID AND MULTI-TOKEN FRAGMENTS

Table 7: LLaDA-8B-Base grid on 500 C4 passages of 200 tokens, five keys; mean and across-key std: the data behind Section 5.2. All rows share the rule filter, keyed selection, gate, and checkerboard schedule and differ only in nucleus p_s, rate ρ, and fragment length L; TPR is measured against an empirical null of 5,000 unwatermarked texts, and the null FPR at $z \geq 4$ is $\leq 4 \times 10^{-4}$ for every row. Default operating point in bold.

Rule	p_s	ρ	Edit %	ΔPPL %	Sem. sim.	Entity	z	TPR@10%	TPR@1%	TPR@0.1%
Gumbel	1.0	0.15	2.57±0.48	6.3±1.9	0.993	0.998	3.30	0.890	0.701±.106	0.479
Gumbel	1.0	0.25	4.30±0.53	11.0±2.1	0.988	0.995	4.30	0.962	0.873±.048	0.725
Gumbel	1.0	0.35	5.85±0.66	16.0±3.1	0.984	0.994	5.10	0.983	0.935±.025	0.854
Gumbel	1.0	0.50	8.62±0.93	26.5±4.3	0.975	0.991	6.25	0.988	0.971±.012	0.945
Gumbel	1.0	0.60	10.68±0.81	35.2±3.9	0.970	0.989	6.92	0.992	0.977±.011	0.962
Gumbel	0.9	0.15	2.43±0.44	3.5±0.9	0.993	0.998	2.49	0.828	0.535±.109	0.275
Gumbel	0.9	0.25	4.07±0.48	6.6±1.2	0.989	0.996	3.26	0.929	0.746±.058	0.496
Gumbel	0.9	0.35	5.52±0.60	9.7±1.8	0.986	0.994	3.84	0.961	0.852±.044	0.686
Gumbel	**0.9**	**0.50**	**8.09**±0.84	**15.9**±2.5	**0.978**	**0.991**	**4.72**	**0.983**	**0.939**±.018	**0.874**
Gumbel	0.9	0.60	10.03±0.72	21.6±2.0	0.973	0.989	5.25	0.991	0.958±.011	0.920
Gumbel, L=2 seq.	0.9	0.25	4.70±1.13	16.0±4.1	0.988	0.994	3.46	0.923	0.770±.126	0.556
Gumbel, L=2 seq.	0.9	0.35	6.44±1.36	22.9±5.5	0.983	0.992	4.07	0.965	0.862±.072	0.756
Gumbel, L=2 seq.	0.9	0.50	9.10±1.25	35.2±5.4	0.976	0.989	4.92	0.981	0.937±.017	0.846

Selection rate, truncation and calibration. Table 7 sweeps $\rho \in \{0.15, \ldots, 0.6\}$ with and without nucleus truncation ($p_s \in \{0.9, 1.0\}$) on the 500-passage C4 benchmark under five keys. Each step of ρ adds $\approx$2 points of edit rate and $\approx$0.5 of z; TPR@1% reaches 0.94 at $\rho = 0.5$ and 0.96–0.98 at $\rho = 0.6$. Removing truncation buys $+1.4$ z at equal ρ, because the tail tokens it restores carry the most surprisal, but at 1.7$\times$ the perplexity cost, so $p_s = 0.9$, $\rho = 0.5$ is the highest-TPR point within the fidelity budget of Section 5.4. Across all 13 configurations and five keys the empirical FPR at $z \geq 4$ on 5,000 null texts lies between 0 and 4×10^{-4}, and the empirical 1% thresholds lie in $[2.06, 2.53]$ against the nominal 2.33 of the moment-matched Gamma null (Figure 7); every TPR in the paper is nevertheless reported against the empirical threshold.

Multi-token fragments. The selection of Section 4.2 generalizes from single positions to fragments of L consecutive tokens: the index axis is cut into blocks $[kL, (k+1)L)$, block k is selected iff $\mathrm{PRF}(\mathsf{k}_{\mathrm{sel}}, kL) < \rho$, ineligible tokens inside a selected block stay visible as context, and the checkerboard assigns blocks to the two passes by the parity of k, so that any two masked blocks of a pass are separated by at least L real tokens. Within a fragment the positions can be refilled in parallel (one forward per pass) or sequentially from left to right (all fragments of a pass advancing together, L forwards per pass). Fragments buy no signal: conditioned on the first refilled token, the second is almost deterministic and is gated out, so at equal ΔPPL the L=2 rows of Table 7 are dominated by L=1. Sequential decoding lowers the per-edit perplexity cost of L=2 but also its signal (Table 16). Longer fragments also interact badly with left-context keys, because the left neighbor inside a fragment is itself being resampled and the detector then re-derives a different seed at that position; position keys are unaffected. L=1 is therefore the default and the only setting used in the main text.

D.2 Detectability across length, source, dLLM and domain

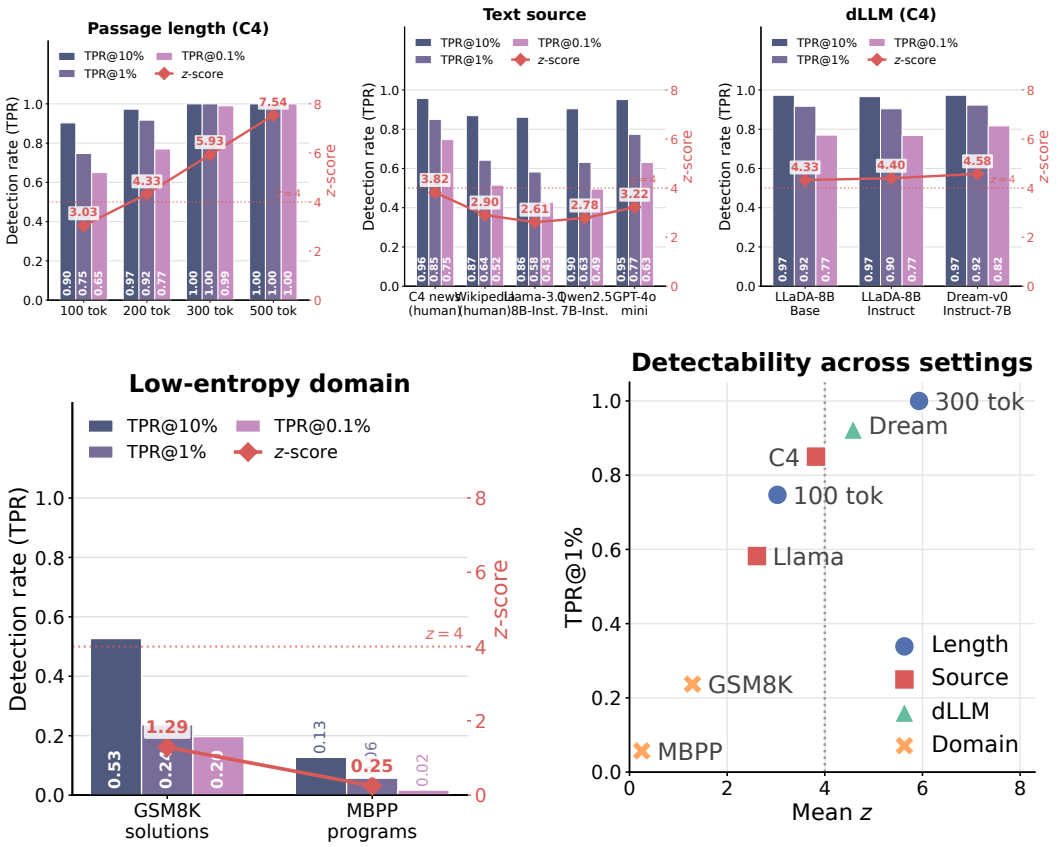


Figure 9: Detectability of ReDiffMark across passage length, text source, dLLM and low-entropy domains: TPR at three FPRs (bars) and mean z (line), and TPR@1% against z over all settings (Figure 2). Numbers in Table 8.

Figure 9 evaluates the same method across passage lengths, text sources, and dLLMs. Length and dLLM comparisons use the default point, while the text-source comparison fixes the lower-edit operating point $\rho = 0.35$ across all sources. Scored positions T_e grow linearly with length while edit rate and ΔPPL do not, so z grows as $\sqrt{T_e}$ and single-passage detection is essentially certain from 300 tokens; 100-token passages, with 12 scored positions, reach 0.75. LLM-generated and Wikipedia text is more predictable to the dLLM, so at equal ρ fewer positions pass the gate and TPR@1% falls from 0.85 to 0.58–0.77, GPT-4o-mini text being the least predictable generator; a target TPR on such a source is reached by moving up the front of Table 7. Dream-v0-Instruct-7B reproduces the default operating point (0.923 vs. 0.917) and LLaDA-8B-Instruct matches it at 45–55% higher ΔPPL, which is why Base is the default. Arithmetic and code expose almost no editable positions (5.6 and 0.4 per passage) and are not detectable; Table 10 shows that 99.0% of the numbers and 99.7% of the final answers of GSM8K solutions survive, whereas on MBPP pass@1 still drops from 1.00 to 0.89, so code should not be watermarked this way at all (Section 6).

Table 8: Detectability of ReDiffMark across passage length, text source and dLLM. T_e = scored positions per passage; Fwd. = forward passes to embed; s/p = embedding seconds per passage (batch 1). ‡Rows whose null holds 300 passages cannot resolve a 0.1% FPR (the threshold would be the largest null score); the 200-token rows use the 1,000-passage C4 null and the text-source block pools five keys (1,500 null texts).

Setting	T_e	z	TPR@10%	TPR@1%	TPR@0.1%	Fwd.	s/p
Passage length (C4, default point, one key, n=300, paired null)							
100 tokens	12.2	3.03	0.903	0.747	—‡	2	0.21
200 tokens	25.3	4.33	0.973	0.917	0.770	2	0.37
300 tokens	45.8	5.93	1.000	1.000	—‡	2	0.60
500 tokens	74.2	7.54	1.000	1.000	—‡	2	0.96
Text source (ρ=0.35, p_s=0.9, five keys, n=300; 0/1,500 null texts at $z\geq4$)							
C4 news (human)	17.7	3.82±0.28	0.957	0.850±.043	0.747		
Wikipedia (human)	13.2	2.90±0.15	0.869	0.642±.044	0.515		
Llama-3.1-8B-Instruct	11.3	2.61±0.22	0.861	0.582±.090	0.428		
Qwen2.5-7B-Instruct	13.3	2.78±0.17	0.904	0.631±.085	0.495		
GPT-4o-mini	16.3	3.22±0.23	0.951	0.774±.077	0.631		
dLLM (C4, default point, one key, n=300)							
LLaDA-8B-Base	25.3	4.33	0.973	0.917	0.770		
LLaDA-8B-Instruct	24.3	4.40	0.966	0.904	0.768		
Dream-v0-Instruct-7B	24.7	4.58	0.973	0.923	0.817		
Low-entropy domain (default point, one key, n=300)							
GSM8K solutions	5.6	1.29	0.527	0.237	—‡		
MBPP programs	0.4	0.25	0.127	0.057	—‡		

Table 9: Text-source generality with fidelity columns: the same hyper-parameters (top-p 0.9, $\rho = 0.35$, $L = 1$) applied to four text sources ($n = 300$ each, five keys; null = unwatermarked passages of the same source, $0/1{,}500$ false positives at $z \geq 4$ for every source). The watermark embeds and is detected on all sources, and the operating point shifts with the entropy the dLLM assigns to the source.

Source	Edit %	ΔPPL %	Sem. sim.	Entity	z	TPR@10%	TPR@1%	TPR@0.1%
Human, C4 news	5.45±0.61	9.2±2.2	0.986	0.994	3.82±0.28	0.957	0.850±.043	0.747
Llama-3.1-8B-Inst. continuation	3.32±0.44	6.3±1.1	0.991	0.996	2.61±0.22	0.861	0.582±.090	0.428
GPT-4o-mini continuation	4.84±0.58	6.2±1.2	0.988	0.990	3.22±0.23	0.951	0.774±.077	0.631
Qwen2.5-7B-Inst. continuation	3.69±0.39	5.8±1.1	0.993	0.996	2.78±0.17	0.904	0.631±.085	0.495
Human, Wikipedia (HotpotQA)	3.87±0.40	5.4±0.7	0.993	0.994	2.90±0.15	0.869	0.642±.044	0.515

Table 10: Passage length (top; C4 passages of the given length, default point, one key, $n = 300$, paired null of the same unwatermarked passages) and low-entropy domains (bottom; task metrics computed by executing the MBPP test lists and by checking that every number and the final answer of the GSM8K solution survive). A null of 300 passages cannot resolve a 0.1% FPR, so TPR@0.1% is reported only for the 200-token row, whose null holds 1,000 passages.

Tokens	Eligible	Selected	T_e	Edit %	ΔPPL %	Sem. sim.	z	$z\geq4$	TPR@10%	TPR@1%	TPR@0.1%	s/passage
100	58.5	23.5	12.2	6.80	10.0	0.970	3.03	0.240	0.903	0.747	—	0.21
200	115.0	49.9	25.3	6.71	11.6	0.982	4.33	0.570	0.973	0.917	0.770	0.37
300	179.2	88.9	45.8	8.35	12.6	0.982	5.93	0.900	1.000	1.000	—	0.60
500	301.9	146.4	74.2	8.12	11.0	0.987	7.54	0.983	1.000	1.000	—	0.96

Domain	Eligible	T_e	Edit %	ΔPPL %	z	TPR@1%	Task metric (original → watermarked)
GSM8K solutions (n=300)	69.9	5.6	1.51	5.8	1.29	0.237	numbers kept 0.990; final answer kept 0.997
MBPP programs (n=300)	8.9	0.4	0.35	1.3	0.25	0.057	pass@1 1.000 → 0.887

D.3 COST BREAKDOWN

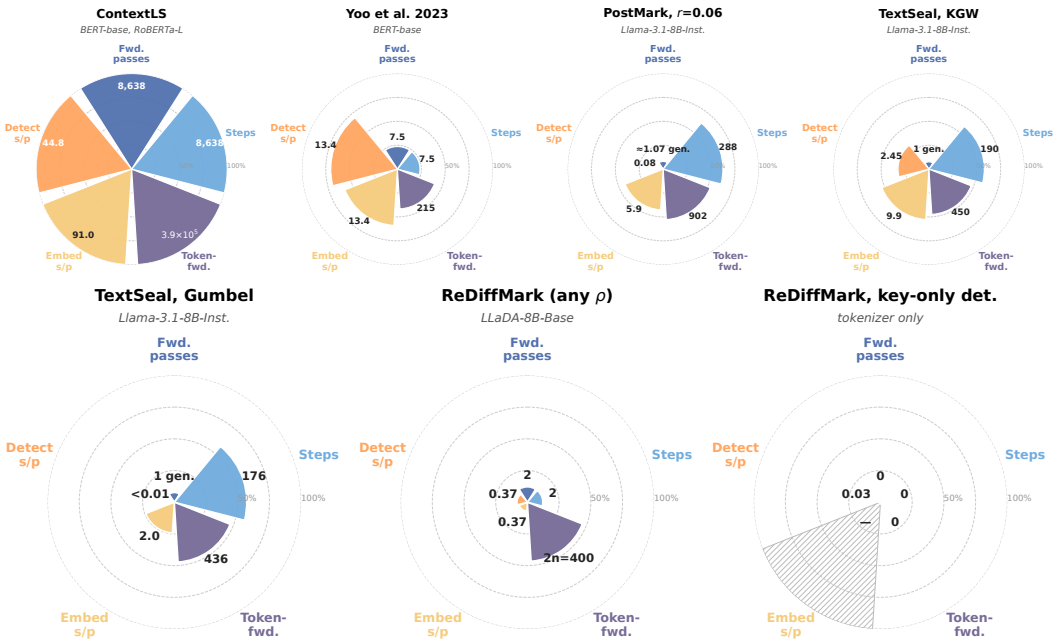


Figure 10: Cost of post-hoc watermarks on 200-token C4 passages (batch 1, one A800-80GB), one polar chart per method; each sector is log-scaled against the largest value of that metric. Numbers in Table 11.

Two normalisations. A forward pass is not a unit of compute: a dLLM forward covers the whole 200-token sequence, whereas one step of autoregressive rewriting is a single-token forward with a KV cache. Counting passes therefore compares *sequential steps* (latency), while counting token-forwards, our FLOPs proxy, compares *arithmetic*: ReDiffMark costs $2n$ token-forwards and an AR rewrite of n tokens costs n decoding steps plus the prefill, i.e. the same order. We report both, and the conclusions differ.

Cost breakdown. Figure 10 and Table 11 decompose the cost of every method: forward passes, sequential steps, token-forwards, and embedding and detection wall-clock at batch 1. Embedding ReDiffMark is two forward passes (0.2–0.4 s) and detection replays exactly the same two; the invariant-feature scheme needs 13 s to embed and 13 s again to detect, of which only 4% is its BERT forward, the rest being candidate re-parsing, and ContextLS 91 s; the rewriting baselines spend 176–190 sequential decoding steps to embed (PostMark 288) and one scoring forward to detect. Wall-clock grows linearly with length for us (0.21/0.37/0.60/0.96 s at 100/200/300/500 tokens, Table 8) because the two forwards attend over more tokens, not because more steps are taken.

Equal latency and equal FLOPs. The left and middle panels of Figure 3 plot TPR@1% against embedding wall-clock and against token-forwards. ReDiffMark is a near-vertical curve traced by $\rho \in \{0.15, \ldots, 0.6\}$, since its cost barely depends on ρ (0.24–0.43 s). To give the rewriting baselines a curve rather than a point they are throttled: TextSeal rewrites only the first k tokens of the passage, $k \in \{25, 50, 100, 150, 200\}$, so that both its cost and its signal scale with k, and PostMark sweeps its insertion ratio $r \in \{0.02, \ldots, 0.12\}$; the MLM-era schemes admit no budget knob and appear as single points. At the iso-latency line of 0.37 s the rewriting baselines can rewrite fewer than 25 tokens and reach less than 0.25; they overtake ReDiffMark only from about 4 s onwards, at which point they have already changed more than 43% of the passage. On the FLOPs axis the two families lie closer: at $\approx 2n$ token-forwards TextSeal-Gumbel reaches 0.98–0.99 against our 0.94, so what equal arithmetic buys is not detection but fidelity, 8% of the tokens edited against 79%. ReDiffMark is also compute-saturated: spending twice the forwards on a second round ($n_{\text{rounds}} = 2$, 0.913 vs. 0.917) or on sequential two-token fragments buys no signal (Table 16), whereas the rewriting baselines' signal grows linearly with budget, so the crossing point is a property of the setting rather than of tuning.

Signal per edited token. The right panel of Figure 3 returns to the question of Section 1, whether a handful of local edits can each carry a verifiable signal. Since the detection statistic's z^2 is proportional to the number of scored positions, z^2 per changed token is an additive measure of the signal one edit carries: at the default point ReDiffMark obtains $4.72^2/16.2 \approx 1.4$ from 16 edited tokens per passage, whereas the full-rewrite watermarks obtain 0.15–0.49 from ≈150. The scheme of Yoo et al. (2023) is competitive on this axis and is balanced by its 13 s per passage and its five-fold perplexity per edited token: it buys its per-edit signal with forced candidates and rule search, we buy ours with two bidirectional forwards.

Table 11: Cost of post-hoc watermarks on 200-token C4 passages (batch 1, one A800-80GB). Token-fwd. counts single-token forward equivalents (FLOPs proxy); Steps counts sequential model calls.

Method	Model	Fwd. passes	Steps	Token-fwd.	Embed s/p	Detect s/p	Mem. (GB)
ContextLS	BERT-base, RoBERTa-L	8,638	8,638	3.9×10^5	91.0	44.8	1.8
Yoo et al. 2023	BERT-base	7.5	7.5	215	13.4	13.4	0.4
PostMark, r=0.06	Llama-3.1-8B-Inst.	≈1.07 gen.	288	902	5.9	0.08	17.5
TextSeal, KGW	Llama-3.1-8B-Inst.	1 gen.	190	450	9.9	2.45	15.2
TextSeal, Gumbel	Llama-3.1-8B-Inst.	1 gen.	176	436	2.0	<0.01	15.1
ReDiffMark (any ρ)	LLaDA-8B-Base	2	2	$2n$=400	0.37±0.02	0.37	18
ReDiffMark, key-only det.	tokenizer only	0	0	0	—	0.03	—

D.4 TEXT FIDELITY

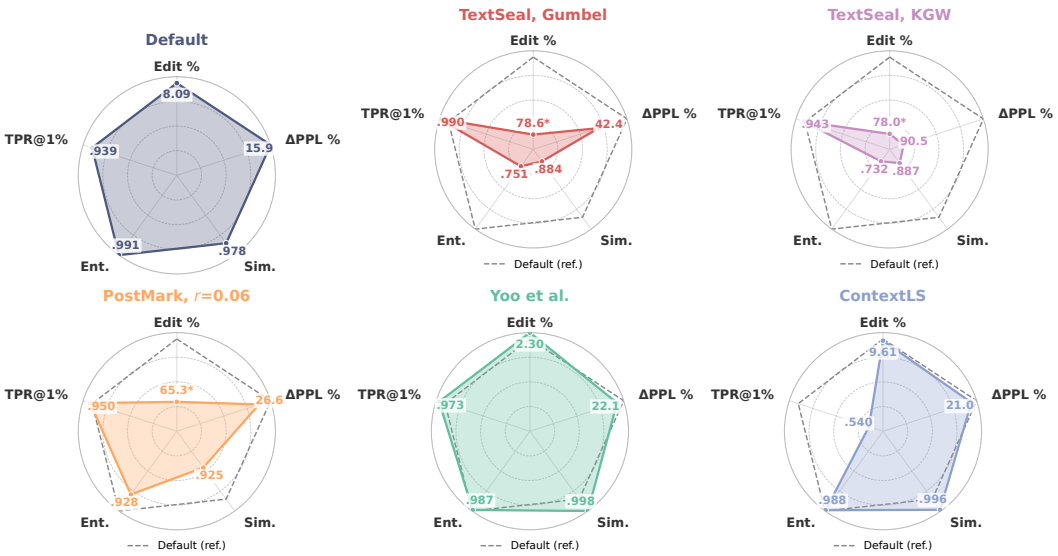


Figure 11: Fidelity against detection on 200-token C4 passages, one radar per method (outward is better; the dashed outline is the ReDiffMark default point). The three of Figure 4 are among them; numbers in Table 12.

Table 12: Text fidelity and quality against detection on 200-token C4 passages; the ReDiffMark row is the five-key mean of Table 1, the baseline rows are single-key runs (n=300); the radars of Figure 11 draw these rows. *edit distance.

Configuration	Edit %	ΔPPL %	Sim.	Ent.	TPR@1%
ReDiffMark (default)	8.09	15.9	0.978	0.991	0.939
TextSeal, Gumbel	78.6*	42.4	0.884	0.751	0.990
TextSeal, KGW	78.0*	90.5	0.887	0.732	0.943
PostMark, r=0.06	65.3*	26.6	0.925	0.928	0.950
Yoo et al.	2.30	22.1	0.998	0.987	0.973
ContextLS	9.61	21.0	0.996	0.988	0.540

Against the baselines. The fidelity columns of Table 1 separate the families. Rephrasing rewrites 78% of the tokens, raises perplexity by 42–91% under one key (the across-key means of Table 1

are dominated by a few degenerate rewrites; the medians are 39–46%) and keeps only 73–75% of capitalised words and numbers. PostMark preserves nothing in place either: at $r = 0.06$ it lengthens the passage by 43% at $\Delta\text{PPL} = 26.6\%$, and at the authors' default $r = 0.12$ by 74% at 49.9% (Appendix A.3). ContextLS changes 9.6% of the tokens once its own detokenisation is discounted (16.1% against the raw original). The scheme of Yoo et al. (2023) changes only 2.3% of the tokens but at five times our perplexity per edited token, because its substitutions are forced choices among BERT candidates that change tense, modality or relation (*was→is*, *will→may*). ReDiffMark's perplexity cost is that of resampling, not of the watermark: at identical positions the keyed Gumbel refill costs 17.5% against 17.9% for plain multinomial resampling and 1.6% for an argmax refill (Appendix B.4), so what we pay is the entropy the dLLM assigns to positions whose right context the scoring LM never sees. Appendix E shows what the edits look like.

D.5 Robustness

Setting. Every attack in this section acts on the 300 C4 passages of 200 tokens of the main text, watermarked at the default point under one key, in both key modes (position keys and left-context keys, Section 4.4). The same attack is applied to the watermarked passages and to their unwatermarked originals, and TPR@1% is read against that *paired* attacked null, whose FPR at $z \geq 4$ is 0 for every attack; "sem. sim." is the sentence similarity between the attacked and the unattacked watermarked passage, i.e. the attacker's own semantic cost. Table 13 lists every attack twice: against the paired null, and against a *fixed* threshold set once on clean unwatermarked text, the empirical 1% and 0.1% quantiles of the 1,000-passage clean C4 null of the same key ($z \geq 2.25$ and $z \geq 3.21$ for position keys, $z \geq 2.27$ and $z \geq 2.69$ for left-context keys), applied unchanged to the attacked watermarked passages (TPR) and to the attacked unwatermarked originals (FPR after attack). The fixed threshold is what a deployed detector uses, since it never sees the attack; the paired null answers whether the attack itself can be told apart. Table 14 runs the same attacks against the post-hoc baselines (authors' code, one key, same passages; PostMark at the $r = 0.06$ operating point of Table 1 with the local Llama-3.1-8B-Instruct inserter) and against the RoBERTa-large control of our own pipeline, which is the ablation of Table 17 rather than a published baseline, with TPR@1% against the paired attacked null of each detector. Random substitution acts on LLaDA tokens and almost always changes the RoBERTa tokenisation length, which desynchronises the control's position keys.

Four edit scenarios. Each scenario acts on a fraction $a \in \{0, 10, 20, 30, 40\}\%$ of the tokens: *deletion* of random tokens; *substitution* by tokens drawn uniformly from the vocabulary; *insertion* of such tokens between existing ones; and *washing*, which re-diffuses a random fraction a of the rule-eligible positions with an *unwatermarked* LLaDA-8B-Base, i.e. the attacker runs our own embedding operation without the key . Panels (a)–(d) of Figure 5 give the full ROC curve of each scenario at every a (TPR against FPR on a linear axis from 0 to 1, read off the paired null) for both key modes; Figure 12 repeats them on a logarithmic FPR axis. The two length-preserving scenarios affect both key modes alike and degrade smoothly: 10% random substitution leaves TPR@1% at 0.59–0.66 while already costing the attacker a similarity of 0.83, and washing, whose fluent refills cost the attacker almost nothing in similarity (0.98 at 30%), still leaves 0.62–0.72 at 30%, i.e. at three times our edit budget; only re-diffusing 80% of the eligible positions, eight times our budget, removes the watermark. Deletion and insertion shift every downstream index, so a single one drives position keys to $z \approx 0$ (0.013 at 20% deletion) regardless of a, whereas left-context keys, keyed on the previous token, lose at most the following position and retain 0.68 at 10% and 0.48 at 20% deletion.

Re-translation. The attacker back-translates the passage (en→de→en with GPT-4o-mini), which rewrites 97% of the tokens at a similarity of 0.95 to the watermarked text. Panel (e) compares TPR@1% after back-translation at every selection rate $\rho \in \{0.15, \ldots, 0.6\}$ with position keys and with left-context keys: position keys retain 0.04–0.05 at every ρ; left-context keys retain 0.18, 0.23, 0.28, 0.36 and 0.30 at $\rho = 0.15$, 0.25, 0.35, 0.5 and 0.6. Owning every token would not help: the rephrasing baselines, keyed on the previous token like our left-context mode, retain 0.44–0.78 after back-translation only because they regenerated the whole passage, and 0.06–0.14 after a free paraphrase (Table 14); a strong-LLM paraphrase leaves left-context keys at 0.13.

What the attacker pays. Panel (f) puts every attack on one axis: the sentence similarity between the attacked and the unattacked watermarked passage against the attack strength a, with back-translation

as a reference line. Washing is nearly free for the attacker (0.98 at 30%) yet needs 80% to succeed; substitution succeeds sooner but at 0.83 by 10%; deletion and insertion are cheap and defeat position keys at any a, which is exactly the gap that left-context keys close; back-translation costs 0.05 of similarity and defeats both modes. The attacks that beat ReDiffMark are therefore either those that rewrite the passage or those that a verbatim-storage pipeline rules out.

Fixed threshold. No attack inflates the false-positive rate: at the 1% threshold the attacked originals are flagged 0.3–2.7% of the time and at the 0.1% threshold 0–1.3% ($n = 300$, so one passage is 0.33%), and no attacked original reaches $z \geq 4$ except one washed passage. The TPR at the fixed 1% threshold stays within 0.06 of the paired-null value for every attack except 50% washing, where the paired null is the stricter of the two (0.47 and 0.42 fixed against 0.36 and 0.29 paired), so the conclusions of this section do not depend on the paired null.

Table 13: Detailed robustness on C4 passages.

		Position keys					Left-context keys				
		fixed 1%		fixed 0.1%		paired	fixed 1%		fixed 0.1%		paired
Attack	Param.	TPR	FPR	TPR	FPR	TPR@1%	TPR	FPR	TPR	FPR	TPR@1%
None	—	0.917	0.027	0.767	0.003	0.860	0.900	0.017	0.840	0.003	0.867
Random deletion	10%	0.017	0.013	0.000	0.000	0.017	0.690	0.017	0.603	0.000	0.680
Random deletion	20%	0.013	0.017	0.000	0.003	0.013	0.487	0.017	0.373	0.000	0.480
Random deletion	30%	0.013	0.017	0.000	0.000	0.000	0.367	0.017	0.240	0.003	0.317
Random deletion	40%	0.017	0.013	0.000	0.000	0.013	0.240	0.010	0.143	0.000	0.257
Random substitution	10%	0.593	0.013	0.370	0.000	0.593	0.707	0.027	0.593	0.010	0.660
Random substitution	20%	0.417	0.020	0.213	0.000	0.380	0.507	0.013	0.363	0.003	0.487
Random substitution	30%	0.257	0.007	0.130	0.000	0.283	0.327	0.007	0.220	0.007	0.350
Random substitution	40%	0.140	0.007	0.040	0.000	0.153	0.187	0.017	0.090	0.003	0.173
Random insertion	10%	0.033	0.013	0.003	0.000	0.033	0.753	0.013	0.690	0.010	0.703
Random insertion	20%	0.013	0.010	0.003	0.000	0.017	0.597	0.017	0.503	0.003	0.587
Random insertion	30%	0.023	0.010	0.000	0.000	0.023	0.427	0.003	0.323	0.000	0.470
Random insertion	40%	0.017	0.017	0.000	0.000	0.017	0.307	0.013	0.200	0.007	0.277
Washing, same dLLM	10%	0.860	0.023	0.687	0.003	0.807	0.847	0.013	0.767	0.000	0.833
Washing, same dLLM	20%	0.817	0.017	0.543	0.000	0.793	0.760	0.023	0.657	0.003	0.710
Washing, same dLLM	30%	0.707	0.007	0.443	0.000	0.720	0.667	0.013	0.570	0.003	0.623
Washing, same dLLM	40%	0.623	0.020	0.370	0.003	0.593	0.553	0.013	0.427	0.003	0.517
Washing, same dLLM	50%	0.467	0.017	0.223	0.003	0.363	0.417	0.017	0.317	0.013	0.293
Washing, same dLLM	80%	0.083	0.020	0.023	0.003	0.057	0.100	0.017	0.040	0.007	0.063
Truncation	keep 50%	0.563	0.013	0.313	0.000	0.540	0.600	0.007	0.483	0.003	0.633
Paraphrase (GPT-4o-mini)	1 hop	0.017	0.020	0.000	0.000	0.007	0.103	0.007	0.050	0.003	0.130
Back-translation	en→de→en	0.043	0.013	0.013	0.000	0.037	0.357	0.010	0.243	0.000	0.363
Chained paraphrase	3 hops	0.017	0.013	0.000	0.000	0.017	0.067	0.007	0.033	0.000	0.073

Table 14: Robustness of the post-hoc baselines under the attacks of Table 13.

Attack	Param.	TextSeal KGW	TextSeal Gumbel	PostMark	Yoo et al.	MLM control
None	—	0.943	0.990	0.950	0.973	0.883
Random deletion	10%	0.817	0.983	0.927	0.190	0.013
Random deletion	20%	0.677	0.973	0.747	0.047	0.003
Random deletion	30%	0.563	0.897	0.570	0.027	0.017
Random deletion	40%	0.357	0.850	0.707	0.027	0.013
Random substitution	10%	0.807	0.983	0.737	0.090	0.050
Random substitution	20%	0.440	0.863	0.613	0.017	0.013
Random substitution	30%	0.403	0.840	0.383	0.013	0.007
Random substitution	40%	0.097	0.583	0.213	0.010	0.003
Random insertion	10%	0.870	0.987	0.930	0.223	0.010
Random insertion	20%	0.753	0.970	0.787	0.070	0.000
Random insertion	30%	0.560	0.927	0.627	0.020	0.020
Random insertion	40%	0.387	0.883	0.463	0.017	0.017
Washing, same dLLM	10%	0.947	0.987	0.887	0.767	0.797
Washing, same dLLM	20%	0.910	0.987	0.877	0.460	0.687
Washing, same dLLM	30%	0.870	0.987	0.807	0.413	0.563
Washing, same dLLM	40%	0.770	0.963	0.833	0.277	0.437
Washing, same dLLM	50%	0.687	0.957	0.583	0.147	0.423
Washing, same dLLM	80%	0.393	0.707	0.250	0.060	0.110
Truncation	keep 50%	0.807	0.980	0.623	0.547	0.600
Paraphrase (GPT-4o-mini)	1 hop	0.060	0.137	0.277	0.013	0.007
Back-translation	en→de→en	0.443	0.783	0.613	0.043	0.053
Chained paraphrase	3 hops	0.060	0.060	0.203	0.017	0.013

ROC curves on a logarithmic FPR axis. The ROC panels of Figure 5 use a linear FPR axis from 0 to 1, which shows the whole curve against the chance diagonal: position keys under deletion and insertion lie on the diagonal at every strength, while every other curve stays above it up to $a = 40\%$. Figure 12 places each of them next to the same curves on a logarithmic axis from 10^{-3} to 1, which resolves the low-FPR regime a detector operates in; the dotted vertical line marks 1% FPR.

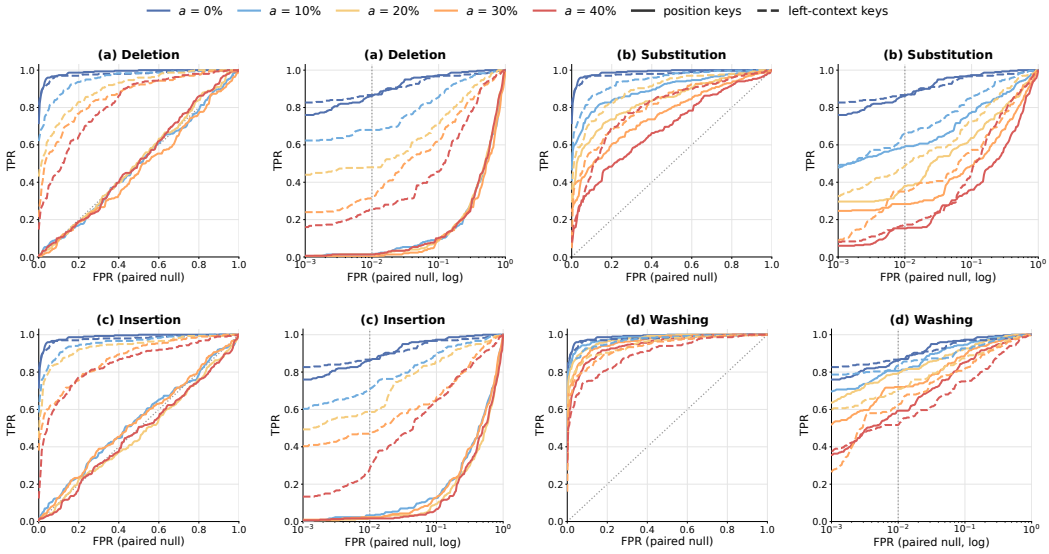


Figure 12: ROC curves of Figure 5 (a)–(d) on the linear FPR axis of the main text (left of each pair) and on a logarithmic FPR axis from 10^{-3} to 1 (right of each pair); 300 C4 passages, default point, paired null, position keys solid and left-context keys dashed.

D.6 Ablations

Figure 6 removes one component at a time from the full method and reports the extra perplexity it costs relative to the full method ($\Delta\Delta$PPL, in points) and TPR at $10/1/0.1\%$ FPR. The embedding ablations (block A) start from the default point on 300 C4 passages under one key; the modulation-rule ablation (block B) keeps positions, gate and checkerboard fixed at $p_s = 1.0$, $\rho = 0.35$ under five

Table 15: Modulation rule at identical positions ($p_s = 1.0$, $\rho = 0.35$, $L = 1$, $n = 500$, five keys, LLaDA-8B-Base): three green-list rules run through the same filter, selection, gate, and checkerboard schedule as ReDiffMark and detected by the green-count z over the same replayed, gated positions. These rows ablate our own scheme; they are not post-hoc baselines.

Rule	Edit %	ΔPPL %	Sem. sim.	Entity	z	TPR@10%	TPR@1%	TPR@0.1%
Gumbel (ReDiffMark)	5.85±0.66	16.0±3.1	0.984	0.994	**5.10**	**0.983**	**0.935**±.025	**0.854**
UMR	5.89±0.65	16.4±2.6	0.984	0.994	1.70	0.644	0.279±.025	0.110
DiPmark	5.88±0.59	16.3±2.1	0.983	0.993	1.70	0.670	0.301±.011	0.147
KGW	6.32±0.68	21.1±3.1	0.983	0.993	2.15	0.806	0.438±.032	0.204

keys on 500 passages and changes only the per-position rule and its detector ($\gamma = 0.5$, green-count z); the detector ablations (block C) reuse the watermarked texts of the full method and change nothing in the embedding.

Checkerboard. Masking all selected positions in one forward lets adjacent masks drift jointly, raising both edits and perplexity (+41 points) for a small gain in TPR. *Keyed selection.* Choosing the top-ρ positions by entropy instead of by key buys $0.92 \to 0.96$ only at 45% more edits, +12 points of ΔPPL and positions an attacker can predict. *Rule filter.* Allowing digits and capitalised words adds $+0.7$ z but drops entity preservation from 0.992 to 0.948. *dLLM infiller.* RoBERTa-large inside the same pipeline leaves the edit-rate–TPR front unchanged but costs 1.3–1.6× the perplexity: keyed selection, the continuous statistic and the replay carry the detectability, the dLLM buys the fluency. *Modulation rule.* At identical positions the unbiased green-list rules leave 78–83% of edited tokens green, which over ≈20 scored positions is $z \approx 2$, the ceiling of Section 4.5. *Likelihood-ratio gate.* Reverting edits whose dLLM log-likelihood drops by more than ε removes the cost together with the signal, because the surprising tokens are the signal. *Replay and weighting.* Skipping the replay (key-only detection, no gate) costs 0.25 of TPR@1%, and surprisal weighting adds $+0.7$ to $+1.1$ z everywhere, most where the signal is weakest.

Table 16: Ablations on 200-token C4 passages, one component changed at a time. Rows are grouped by the run they come from, so that every row compares to the reference row of its own group: the five-key grid ($n = 500$), the single-key ablation round at the default point ($n = 300$, C4 null of 1,000 passages), and the quick single-key rounds on LLaDA-8B-Instruct ($n = 200$). The low-edit point is $\rho = 0.35$ with top-p 0.9; parameters of every variant are those of Table 2. Detection rows re-score the reference embedding and change nothing in the text. ΔPPL is not measured for detection-only rows.

Variant	Edit %	z	TPR@1%	ΔPPL %	Effect
Five-key grid (LLaDA-8B-Base, $n = 500$); reference: low-edit point					
Reference (low-edit point)	5.52	3.84	0.852	9.7	—
Default point	8.09	4.72	0.939	15.9	operating point of the paper
No nucleus truncation	5.85	5.10	0.935	16.0	more signal per edit at more ΔPPL
LLaDA-Instruct backend	5.65	4.03	0.854	14.0	+45% ΔPPL at equal TPR
Two-token fragments, sequential	6.44	4.07	0.862	22.9	dominated by $L = 1$; twice the forwards
Unweighted statistic (detection)	—	3.10	0.722	—	weighting adds +13 points
Unweighted statistic, default point (detection)	—	3.82	0.876	—	weighting adds +6 points
Key-only detection, default point (detection)	—	3.06	0.688	—	replay adds +25 points
Ablation round (LLaDA-8B-Base, one key, $n = 300$); reference: default point					
Reference (default point)	6.71	4.33	0.917	11.6	—
No checkerboard	10.15	6.03	0.960	52.8	adjacent masks drift jointly
Entropy-based selection, no key	9.74	5.44	0.963	23.4	more edits, predictable positions
No candidate-set gate	6.91	4.44	0.927	10.8	no measurable change
No rule filter	7.73	5.03	0.960	17.2	entity preservation $0.992 \to 0.948$
Two rounds of re-diffusion	8.01	4.51	0.913	12.0	re-sampling does not accumulate signal
Three rounds of re-diffusion	8.47	4.53	0.920	11.9	idem
Dream backend	6.67	4.58	0.923	11.9	transfers across dLLM families
Quick rounds (LLaDA-8B-Instruct, one key, $n = 200$); reference: low-edit point					
Reference	6.1	4.35	0.865	16.5	—
Likelihood-ratio gate	4.51	2.57	0.560	6.4	signal and cost are co-located
Lower entropy gate	6.41	4.36	0.865	16.3	no change
Two-token fragments, parallel	7.98	4.92	0.885	36.9	sequential decoding lowers cost and signal
Two-token fragments, sequential	7.05	4.28	0.840	27.1	
Exact likelihood-ratio statistic (detection)	—	2.63	0.870	—	heavy-tailed weights; worse
Detector entropy gate swept (detection)	—	±0.1	±0.01	—	plateau on $[0.3, 0.5]$

Table 17: Ablation of ReDiffMark: extra perplexity relative to the full method and detection at three FPRs. A: default point, one key, n=300 (full method ΔPPL 11.6%, z 4.33; †five keys, n=500, against the five-key default point, ΔPPL 15.9%, TPR 0.983/0.939/0.874). B: identical positions at the low-edit point, five keys, n=500 (full method ΔPPL 16.0%, z 5.10). C: the watermarked texts of the five-key default point of Table 1, re-scored (ΔPPL 15.9%, z 4.72). Parameters of every variant are those of Table 2.

Variant	ΔΔPPL (pt)	TPR@10%	TPR@1%	TPR@0.1%
A. Embedding				
Full method	0	0.973	0.917	0.770
No checkerboard	+41.2	0.987	0.960	0.923
Entropy-based selection, no key	+11.8	0.997	0.963	0.943
No rule filter	+5.6	0.993	0.960	0.887
RoBERTa infiller†	+10.2	0.977	0.924	0.815
Likelihood-ratio gate	−5.2	0.830	0.560	0.425
B. Modulation rule				
Gumbel (full method)	0	0.983	0.935±.025	0.854
KGW	+5.1	0.806	0.438±.032	0.204
DiPmark	+0.3	0.670	0.301±.011	0.147
UMR	+0.4	0.644	0.279±.025	0.110
C. Detector				
Full method	0	0.983	0.939	0.874
Key-only detection, no replay	0	0.905	0.688	0.544
Unweighted statistic	0	0.964	0.876	0.734

D.7 Multi-bit messages

Multi-bit messages. Figure 13 plots, for five ways of coding an 8-bit message into the same embedding (300 C4 passages of 200 tokens, LLaDA-8B-Base, top-p 0.9, $\rho = 0.5$, one key, ≈25 scored positions), the fraction of messages recovered exactly (horizontal) against the presence TPR@1% of the decoded-message statistic (vertical); only the per-position symbol alphabet M and the decoder differ between the points, the edited tokens are identical. The two axes pull against each other: a larger alphabet carries more bits per position, but the presence test then takes the maximum over more candidate messages and its null rises. One bit per position (Yoo et al., 2024; Wang et al., 2024) recovers the message only 21% of the time; a 16-ary allocation recovers it 71% of the time at a presence TPR@1% of 0.69 and is the compromise we recommend; treating the whole message as one symbol (Fernandez et al., 2023) recovers 82% but leaves presence detection at 0.51. At 300 tokens the 16-ary code reaches 91%. The full grid over codes, message lengths and ρ is Table 19 below.

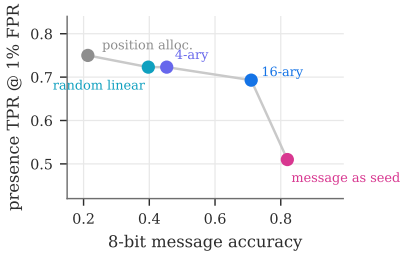


Figure 13: Payload accuracy against presence detection for five message codes.

Multi-source provenance. Table 18 attributes 400-token documents that alternate two Wikipedia sources, each watermarked under its own key at the default point, with segments of 25 to 200 tokens; the candidate set holds the two true keys and two distractors. "Token acc." is the fraction of tokens assigned to the right source by the Viterbi decoder of Appendix D.8; "Doc. rec." and "Doc. prec." are recall and precision of the true sources at the 1% threshold of the per-key z; "False act." is the fraction of distractor keys above that threshold. Precision is ≥0.96 at every segment length and no distractor reaches $z \geq 4$; recall stays near 0.75 because each source owns only half of the scored positions; segmentation accuracy rises from 0.59 for 25-token segments to 0.81 for 200-token ones, and the Bonferroni-corrected false activation is 0 up to 50 candidate keys.

Table 18: Multi-source provenance against the segment length.

Seg. len.	n	Edit %	Token acc.	Doc. rec.	Doc. prec.	False act.
25	108	7.3	0.586	0.736	0.967	0.009
50	118	6.9	0.647	0.758	0.964	0.051
100	133	6.1	0.716	0.748	0.956	0.038
200	130	5.4	**0.812**	0.746	**0.988**	**0.008**

Detection without the model. A detector that holds only the key and the tokenizer recomputes the selection and scores it without replay, gate or weighting: it reaches $\mathrm{TPR@1\%} = 0.688$ against 0.939 for the replay detector, at 0.03 s per passage on a CPU (Figure 10, Table 17), which makes it a usable first-pass filter in front of the two-forward detector.

Coding study. Table 19 is the full grid behind Figure 13: five codes, messages of $m = 8$ and 16 bits, and $\rho \in \{0.35, 0.5\}$, on 300 C4 passages of 200 tokens (LLaDA-8B-Base, top-p 0.9, key 0) against a null of 500 unwatermarked passages. All codes share one embedding pipeline (a hash-seeded Gumbel field, one symbol per gated position), so the edit rate and the number of scored positions T are the same for every code at a given ρ; the codes differ only in how the T symbols are read: *position allocation* assigns one message bit to each position, the 4-ary and 16-ary allocations assign two and four bits, the *random linear code* adds error correction, and *message as seed* uses the whole message as the key of the Gumbel field. Decoding is exhaustive maximum likelihood over the 2^m candidates with the surprisal-weighted score. "Msg. acc." is the fraction of messages recovered exactly and "Bits/passage" $= m\times$ Msg. acc. is the effective payload; "Null z" is the decoded-message statistic on unwatermarked text, inflated by the maximum over the 2^m candidates, and $\mathrm{TPR@1\%}$ is measured against that null. Three things stand out (bold): at $\rho = 0.5$ the 16-ary code is the best compromise for 8 bits (0.71 recovered at presence 0.69), message-as-seed recovers the most messages (0.82) at the worst presence rate (0.51), and no code recovers a 16-bit message reliably at 200 tokens (at most 0.42), because the null grows with the message space.

Table 19: Multi-bit coding study: message recovery and presence detection for five codes, two message lengths and two selection rates.

ρ	Code	m	Edit %	T	Bit acc.	Msg. acc.	Bits/passage	Null z	TPR@1%
0.35	Position allocation	8	4.77	17.3	0.747	0.090	0.72	1.11	0.597
0.35	Position allocation	16	4.75	17.3	0.658	0.000	0.00	1.25	0.547
0.35	4-ary allocation	8	4.85	17.4	0.802	0.273	2.19	1.76	0.490
0.35	4-ary allocation	16	4.88	17.2	0.720	0.030	0.48	2.13	0.403
0.35	16-ary allocation	8	4.77	17.2	0.835	0.497	3.97	2.43	0.420
0.35	16-ary allocation	16	4.74	17.3	0.704	0.040	0.64	3.17	0.257
0.35	Random linear code	8	4.79	17.2	0.629	0.267	2.13	1.27	0.490
0.35	Random linear code	16	4.73	17.2	0.518	0.007	0.11	1.43	0.453
0.35	Message as seed	8	4.69	17.4	0.816	0.633	5.07	2.94	0.320
0.35	Message as seed	16	4.76	17.3	0.624	0.230	3.68	4.62	0.067
0.5	Position allocation	8	6.88	24.8	0.809	0.213	1.71	1.21	0.750
0.5	Position allocation	16	6.83	24.8	0.707	0.007	0.11	1.42	0.677
0.5	4-ary allocation	8	6.98	24.8	0.864	0.453	3.63	1.80	0.723
0.5	4-ary allocation	16	6.97	24.9	0.769	0.050	0.80	2.25	0.590
0.5	16-ary allocation	8	6.76	25.0	0.909	**0.710**	5.68	2.46	**0.693**
0.5	16-ary allocation	16	6.80	25.0	0.778	0.157	2.51	3.26	0.507
0.5	Random linear code	8	6.84	24.9	0.691	0.397	3.17	1.41	0.723
0.5	Random linear code	16	6.80	24.8	0.501	0.030	0.48	1.65	0.670
0.5	Message as seed	8	6.84	25.0	0.912	**0.820**	6.56	2.93	**0.510**
0.5	Message as seed	16	6.83	24.8	0.719	**0.423**	6.77	4.57	0.197

300-token passages. On the 300-token C4 passages of Table 10 ($\rho = 0.35$, ≈33 scored positions, paired null of 300 unwatermarked passages) the message-as-seed code recovers 8-, 12-, and 16-bit messages with accuracy 0.913, 0.760, and 0.587 (bit accuracy 0.95, 0.88, 0.79), and 16-ary allocation with 0.790, 0.463, and 0.300; the presence $\mathrm{TPR@1\%}$ of the decoded-message statistic is 0.74/0.55/0.35 (seed) and 0.80/0.71/0.62 (16-ary). An 8-bit payload is therefore reliable at 300 tokens, and 12 bits at 400–500.

Symbol field. For the coding study the Gumbel uniforms are produced by a keyed 64-bit mixing function (splitmix64) of $(\mathrm{PRF}(\mathsf{k_g}, i),\ s,\ v)$ rather than by a seeded PRNG vector, so that the uniform of a single (position, symbol, token) triple costs $O(1)$ and the detector can score 2^{16} symbols per position; the zero-bit results are unchanged (same edit rates, z, and TPR as with the PRNG field).

D.8 Multi-source provenance

The provenance decoder is a Viterbi pass over states $\mathcal{C} \cup \{\varnothing\}$. At scored position j the emission of state $s \in \mathcal{C}$ is $\prod_{k\in\mathcal{C}} f_k(\mathrm{sc}_{j,k})^{w_j}$ with f_k the density of $\mathrm{Exp}(\text{mean } 1{+}\delta)$ if $k = s$ and $\mathrm{Exp}(1)$

otherwise, where $\mathrm{sc}_{j,k} = -\log(1-u_j^{(k)})$ is the Gumbel score of key k; the null state uses $\mathrm{Exp}(1)$ for every key. Transitions keep the current state with probability $\sigma = 0.97$ and move uniformly otherwise; the initial distribution puts mass 0.5 on the null state. Per-key z values are moment-matched-Gamma z scores of the weighted score sums over the positions assigned to the key (document-level) or over all positions (global). A first attempt at $\rho = 0.35$ on 200-token documents (3–3.6% edits) gave true-source global z of only 1.4–1.8 and chance-level token accuracy, which motivated the 400-token, $\rho = 0.5$ setting of Table 18. The candidate-set size barely matters: pooling all segment lengths ($n \approx 490$ documents), the fraction of distractor keys with global $z \geq 4$ is 0 for $|\mathcal{C}| \in \{2, 5\}$, 0.002 for 10 and 0.004 for 50 keys before correction and 0 after Bonferroni correction, while true-source recall at $z \geq 4$ stays at 0.36–0.47; unwatermarked documents score $z = 0.52$–0.63 under the true keys with no false positive at $z \geq 4$.

E Case Studies

Each card merges the watermarked passage into the original: a ~~struck-out red~~ token is what the original had and the **bold blue** token beside it is what ReDiffMark wrote; everything else is verbatim. The badges give the edited tokens, the sentence similarity to the original, and the detection statistic of that passage before and after watermarking, all at the default point under one key.

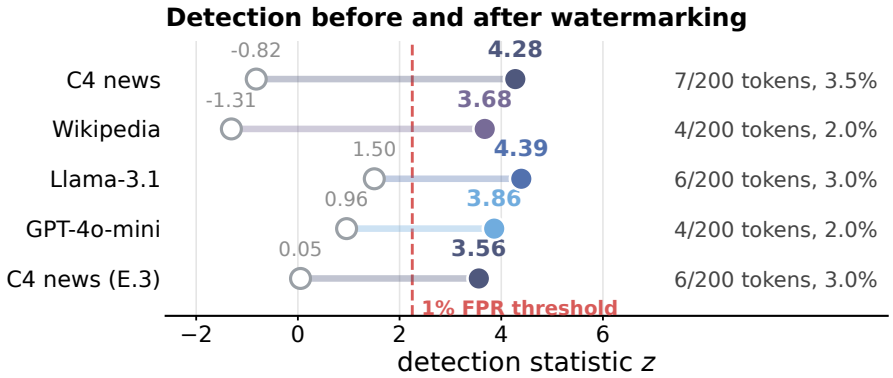


Figure 14: **Every case crosses the threshold.** Detection statistic of each shown passage before (hollow) and after (filled) watermarking, with the edited tokens beside it.

E.1 Representative examples

Human-written news — **C4 `realnewslike`**

Occasional diarrhea is a common occurrence. Most people will experience an episode of diarrhea at least once or twice a year that will ~~disappear~~ **pass** in a couple of days. Luckily, there are many foods to eat that may help a person reduce the symptoms of diarrhea. There are also some foods to avoid when dealing with a bout of diarrhea, and some ~~additional~~ **simple** home care ~~tips~~ **treatments** to consider. Anyone who is experiencing persistent diarrhea should see a doctor, as a person may become dehydrated over time. Diarrhea is a bowel movement that is more liquid than solid or has a loose texture. It is a common ~~problem~~ **condition** and may occur a couple of times each year. Diarrhea will usually last no more than 3 days. Soft, bland foods, such as toast, ~~will~~ **may** help to absorb excess water and ~~ease~~ **reduce** the symptoms of diarrhea. People with chronic digestive conditions, such as irritable bowel syndrome (IBS) or Crohn's disease, may experience diarrhea more ~~regularly.~~ **often.** A person

7 of 200 tokens edited, 3.5% | Similarity 0.996 | z: $-0.82 \rightarrow 4.28$

Entity-heavy encyclopaedic text — **Wikipedia**

Bernhard Bötel (1883–1953) was a German operatic tenor and actor who ~~had~~ **enjoyed** an active career in Germany and Austria during the first half of the 20th century. He made recordings for several record labels ~~during~~ **in** the early years of the recording ~~industry,~~ **era,** including His Master's Voice, Odeon Records, Polydor Records, Tri-Ergon, and Vox Records. On the stage he sang a variety of roles in operas and operettas from leading parts to comprimario roles. His ~~stage~~ **own** repertoire included Belmonte in Wolfgang Amadeus Mozart's "Die Entführung aus dem Serail", Chapelou in Adolphe Adam's "Le postillon de Lonjumeau", Count Almaviva in Gioachino Rossini's "The Barber of Seville", Daniel in Franz Lehár's "The Merry Widow", the Duke of Mantua in Giuseppe Verdi's "Rigo

4 of 200 tokens edited, 2.0% | Similarity 0.999 | z: $-1.31 \rightarrow 3.68$

E.2 Examples across text sources

LLM-written text — **Llama-3.1-8B-Instruct**

Buying ~~advertising~~ **ads** can be expensive and creating ~~the~~ **effective** ads can also be a time-consuming and difficult process. While there's no doubt that advertising and other marketing efforts are ~~crucial~~ **essential** for businesses to reach their target audience and stay competitive, the high costs and complexities involved can be a significant obstacle for many companies. The process of creating effective ads, whether it's through traditional media such as television or radio, or digital platforms like social media and online ~~search,~~ **banners,** requires a great deal of creativity, expertise, and resources. From concept development to scriptwriting, filming, and editing, the production process can be time-consuming and labor-intensive. This is especially true for small businesses or startups that may not have the necessary budget or in-house talent to ~~handle~~ **manage** the task. Even larger companies often find themselves juggling multiple ad campaigns, each with ~~its~~ **their** own unique messaging, target audience, and distribution channels, which can lead to logistical challenges and increased costs. Furthermore, the ever-changing landscape of consumer behavior and technology trends can make

6 of 200 tokens edited, 3.0% | Similarity 0.993 | z: $1.50 \rightarrow 4.39$

LLM-written text — **GPT-4o-mini**

Longboard Media is celebrating its two year anniversary and has revealed some amazing growth ~~numbers,~~ **metrics,** crystalizing its position as the largest ad network in the longboarding niche. In just two years, the company has expanded its reach significantly, ~~boasting~~ **forging** partnerships with over 500 brands and influencers that resonate with the longboarding community. The network has grown exponentially, with monthly impressions climbing to over 50 million, a remarkable increase from the initial ~~launch~~ **monthly** figures. This surge in ~~visibility~~ **reach** has not only solidified Longboard Media's role as a critical player in the advertising landscape but has also provided its clients with unparalleled opportunities to engage with a dedicated audience. The company attributes its success to a combination of innovative marketing strategies and a deep understanding of the longboarding culture. By focusing on content that genuinely resonates with the community, Longboard Media has transformed traditional advertising into a more authentic experience. Brands are no longer merely placing ads; they are becoming part of the longboarding narrative, engaging with riders through storytelling and

4 of 200 tokens edited, 2.0% | Similarity 0.997 | z: $0.96 \rightarrow 3.86$

E.3 QUALITATIVE COMPARISON WITH BASELINES

Original passage — **C4 news, 200 tokens**

Following in her mother's footsteps, 23-year-old Shakisa Olinda Harvey is among the newest admissions to the bar after graduating last Saturday from the Hugh Wooding Law School as an Attorney-at-Law. The bubbly Harvey is the daughter of recently-appointed High Court Judge Justice Simone Morris-Ramlall. Always certain of the career path she wanted to pursue, Harvey, who was called to the bar on Monday, said she has proven that determination, hard work and dedication truly pays off. Presenting her petition before Madame Chief Justice (ag) Roxane George, attorney Abiola Wong-Inniss, to whose chambers Harvey is attached . . .

ReDiffMark — **6 of 200 tokens edited, 3.0%**

Following in her mother's footsteps, 23-year-old Shakisa Olinda Harvey is among the newest admissions to the bar after graduating last Saturday from the Hugh Wooding Law School as an Attorney-at-Law. The bubbly Harvey is the daughter of recently-appointed High Court Judge Justice Simone Morris-Ramlall. Always ~~certain~~ **sure** of the career path she wanted to pursue, Harvey, who was ~~called~~ **admitted** to the bar on Monday, said she has proven that determination, hard work and dedication truly pays off. Presenting her petition before Madame Chief Justice (ag) Roxane George, attorney Abiola Wong-Inniss, to whose chambers Harvey is attached as an associate, described her as industrious. Recalling Harvey's in-service stints . . .

Yoo et al., 2023 — sparse substitution — **5 of 147 words changed**

Following in her mother's footsteps, 23-year-old Shakisa Olinda Harvey is among the newest admissions to the bar after graduating last Saturday from the Hugh Wooding Law School as an Attorney-at-Law. The bubbly Harvey is the daughter ~~of~~ **with** recently-appointed High Court Judge Justice Simone Morris-Ramlall. Always certain of the career path she wanted to pursue, Harvey, who ~~was~~ **got** called to the bar on Monday, said she has proven that determination, hard work and dedication truly pays off. Presenting her petition before Madame Chief Justice (ag) Roxane George, attorney Abiola Wong-Inniss, to whose chambers Harvey ~~is~~ **became** attached as an associate, described her as industrious. Recalling Harvey's in-service stints spent . . .

TextSeal, Gumbel — full rewrite — **keeps 41 of 147 original words**

Shakisa Olinda Harvey, a 23-year-old and the latest in a line of legal professionals, has become an Attorney-at-Law after graduating last Saturday from the Hugh Wooding Law School. Harvey, whose mother is High Court Judge Justice Simone Morris-Ramlall, is joining the ranks of the practicing bar. The driven Harvey, who was called to the bar . . .