TANGO: Watermarking Masked Diffusion Language Models in Token Pairs

Anonymous authors
Paper under double-blind review

Abstract

Masked-diffusion language models fill in masked positions in parallel and in no fixed order. Most practical text watermarks assume left-to-right generation. They key each token to the tokens before it, and in a diffusion model those tokens may still be masked. A fixed green list needs no such context, but it favors the same tokens at every position, so these tokens appear more often in watermarked text. An attacker who compares token frequencies in watermarked and unwatermarked text can recover the list and forge text that the provider's own detector accepts. We present **TANGO**, a watermark for masked-diffusion language models that keys each new token to a nearby token that is already unmasked. A secret key splits the vocabulary into color classes, and TANGO biases the new token toward a color determined by the key and the nearby token's color. The watermark is therefore embedded in *pairs* of tokens. Because the favored color changes from position to position, token frequencies stay close to those of unwatermarked text. Detection needs only the text and the key, and it does not assume any unmasking order. On two masked-diffusion models, TANGO detects nearly all unedited watermarked texts and most edited ones, and frequency attacks that forge the fixed green list fail against it.[1]

1 Introduction

Language models now write text that readers cannot tell apart from human writing, and their providers are increasingly asked to identify which texts their models produced. Text watermarking addresses this need. The provider embeds a secret statistical signal during generation and later tests for its presence. A prominent method is the context-hashed green list (Kirchenbauer et al., 2023), which derives its favored token set from a hash of the preceding tokens. In left-to-right generation, these tokens are already available when the next token is sampled. Masked-diffusion language models (Austin et al., 2021; Sahoo et al., 2024; Nie et al., 2025; Ye et al., 2025) instead fill positions out of order through a *denoising loop*. Under confidence-based decoding, the model unmasks the positions it is most confident about first, regardless of their order in the sequence. When a position is unmasked, the tokens before it may still be masked, so the context-hashed green list cannot be computed.

Other watermarks run on diffusion models but face different problems. Gumbel-max sampling (Aaronson & Kirchner, 2023), which we call the Gumbel rule, can be applied at every denoising step (Bagchi et al., 2025). However, when we key it on the preceding token, as in its autoregressive form, detection is weak in our experiments. A confidence-ordered sampler unmasks a position only once the model is nearly certain of its token. With so little uncertainty left, the Gumbel rule almost always picks that same token, so it embeds little signal (Section 4.2). The fixed green list of Zhao et al. (2024), which we call the red–green list, splits the vocabulary once into green and red tokens and favors the green ones. It needs no context and runs unchanged on diffusion models. However, it favors the same tokens at every position, so an attacker can estimate the green list from token frequencies in watermarked text. With the green list, the attacker can forge text that the detector accepts (Jovanović et al., 2024) or remove the watermark (Zhang et al., 2024b). A provider that needs to prevent forgery therefore cannot use the red–green list.

[1] Anonymized code: `https://anonymous.4open.science/r/tango-watermark`.

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.

We present **TANGO**, a watermark that is designed for the denoising loop and does not rely on a left-to-right generation order. A secret key colors the vocabulary into q classes. TANGO pairs each position with the token at a fixed offset, which we call the position's *tap*. If the tap has already been generated when a position is unmasked, TANGO biases the logits toward the one class whose color satisfies a keyed modular relation with the tap's color. We call this relation the *checksum*. The detector recolors a candidate text with the key and counts the positions where the checksum holds. Because the classes have near-equal size, the checksum holds at about $1/q$ of the positions in unwatermarked text, and the detector flags text where it holds much more often. Detection needs only the text and the key, not the prompt or the model.

The watermark signal is therefore *second order*. It comes from pairs of token colors, not from any single token. Because the classes have near-equal size, the biases at different positions also cancel out on average. If the favored class is uniform and independent across positions, every token appears as often as it would without the watermark (Theorem 1). Ranking tokens by how much more often they appear in watermarked text therefore reveals no difference between the color classes. Empirically, this attack recovers and forges the red–green list but stays at chance against TANGO, and none of its forgeries is accepted (Section 4.3). An attacker who knows TANGO's design can instead count token pairs. At every forging bias and every number of texts we test, these forgeries succeed less often against TANGO than forgeries of the red–green list succeed against it.

Embedding the signal in token pairs raises two concerns. First, editing either token of a pair can break it. TANGO therefore assigns tokens to classes by *meaning*. It sorts tokens by a keyed projection of their embeddings and cuts the sorted list into equal-sized classes. Tokens with similar meanings then tend to share a class, so the checksum often survives synonym substitution attacks. Second, forcing a relation between neighbors could hurt the quality of the generated text. TANGO favors a whole class of tokens, so the model still chooses freely within that class, and quality degrades gradually as the bias grows. Under synonym substitution and word insertion, TANGO detects nearly as many texts as the red–green list. TANGO is more sensitive to word deletion than the red–green list. Deleting a word breaks every pair that spans it, whereas the red–green list scores each token independently (Section 4.2).

Contributions.

- We introduce TANGO, a coloring-based watermark for masked-diffusion language models, which couples the color of each token to the color of one already-unmasked neighbor (Section 3).
- We prove three properties of TANGO (Section 3.4). The expected detection score grows as $\sqrt{T}$, where T is the number of scored pairs. When the colors are balanced and the favored class is uniform and independent across positions, the watermark leaves the expected frequency of every token unchanged, so token frequencies do not reveal the key. Replacing tokens at random with rate ρ multiplies the expected detection score by $(1-\rho)^r$, where r is the number of tokens a checksum reads, so checksums over fewer tokens are more robust to edits.
- On two masked-diffusion models, TANGO detects nearly all unedited watermarked texts and most edited ones. Its detection rate on unedited text and its perplexity are comparable to those of the red–green list. At similar perplexity, TANGO detects far more texts than the Gumbel rule (Section 4.2). TANGO is also harder to forge, since frequency attacks that forge the red–green list fail against it (Section 4.3).

2 Preliminaries

Decoding in masked-diffusion language models. A masked-diffusion model starts from a block of masks and, at each step, chooses which masked positions to fill and what to write there. Let $\mathcal{V}$ be the vocabulary, $V = |\mathcal{V}|$, and $\mathsf{m} \notin \mathcal{V}$ the mask symbol. Given a prompt, the model fills a length-L block over S denoising steps, producing states $x^{(0)}, \ldots, x^{(S)} \in (\mathcal{V} \cup \{\mathsf{m}\})^L$ with $x^{(0)}$ fully masked and $x^{(S)} = (t_1, \ldots, t_L)$ the output. At each step j, one bidirectional forward pass of the model (parameters θ) yields marginals $p_\theta(x_i = v \mid x^{(j-1)})$ for every masked position. The step samples a candidate at each masked position and keeps the K most confident positions, re-masking the rest

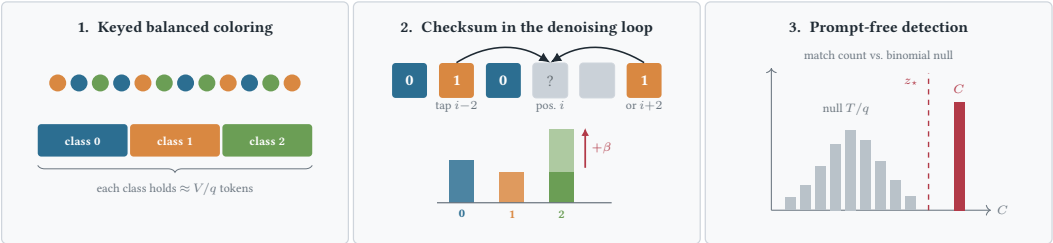

Figure 1: **How TANGO works. (1)** A secret key sorts the vocabulary by a keyed embedding score and cuts it into q color classes of near-equal size, so that tokens with similar embeddings tend to share a color. **(2)** When the tap of a masked position is already unmasked, a logit bias $+\beta$ favors the one class that satisfies the checksum with the tap's color. The favored class changes from position to position, so when the tap colors are evenly spread, no token is favored on average. **(3)** Detection counts the matches C among the T scored positions and flags the text when C exceeds the $\mathrm{Binom}(T, 1/q)$ null by more than a threshold $z_\star$ standard deviations.

(Nie et al., 2025; Ye et al., 2025). In both LLaDA and Dream, the confidence of a candidate is the probability the model assigns to the sampled token, and the step's schedule sets K.[2] Unmasking is irreversible, so the order in which positions are filled is a permutation of $\{1, \ldots, L\}$ that depends on the data.

Requirements. This generation setting motivates three requirements. *(R1) Prompt-free detection.* The detector should use only the candidate text and the secret key. *(R2) Order-agnostic detection.* Detection should not need the unmasking order, which the final text does not reveal. *(R3) Frequency-hidden key.* The token frequencies of watermarked text should not make the secret key easy to infer.

Attacker. We assume the attacker knows the watermarking method and every design parameter, but not the secret key. It can collect watermarked and unwatermarked texts, edit a watermarked text, and generate text with a language model it controls. Its goal is either to forge, that is, to produce text that the provider's detector accepts, or to remove the watermark from a text by editing it.

3 TANGO

Overview. Before generation, the secret key k pseudorandomly assigns every vocabulary token one of q colors through a coloring $\chi : \mathcal{V} \to \mathbb{Z}_q$ that is balanced (Section 3.1), where $\mathbb{Z}_q = \{0, \ldots, q-1\}$ with arithmetic modulo q. During generation, TANGO considers each masked position i whose tap, the position $i - \delta$ at a small fixed lag δ, is already unmasked. It adds a bias $\beta > 0$ to the logits of every token of the one color that satisfies the checksum with the tap's color. The step then selects which candidates to unmask by the sampler's usual rule, keeping the most confident ones, with confidences computed from the biased distribution. Detection needs neither the model nor the prompt. It recolors the text with χ, counts the positions where the checksum holds, and compares the count with the binomial distribution expected by chance (Figure 1).

3.1 Balanced colorings

A coloring is *balanced* if every class contains either $\lfloor V/q \rfloor$ or $\lceil V/q \rceil$ tokens. Balance makes plausible the uniform-color hypothesis under which an unwatermarked position satisfies the checksum with probability $1/q$ (Proposition 1), and it is the token-count counterpart of the balance of class probabilities that Theorem 1 assumes. How tokens are assigned to classes determines TANGO's robustness to edits.

We consider three ways of assigning tokens to colors. A **hash** coloring partitions a keyed pseudo-random permutation of the vocabulary into q equal blocks. It ignores the meaning of the token, so replacing a token by a synonym randomizes its color, and the pair it belongs to no longer satisfies

[2] Semi-autoregressive variants decode blocks left to right but remain order-free within a block.

the checksum more often than chance. A **semantic-cluster** coloring is a balanced clustering of the token embeddings. Tokens with similar embeddings, which often occur near each other in text, then share a color, so unwatermarked text satisfies the checksum more often than chance. At small q this raises the false-positive rate far above its nominal value (60% at $q = 2$ and 30.5% at $q = 3$, Section E.1). We propose **semantic-quantile** coloring, which sorts tokens by the projection $\langle \mathbf{e}_v/\|\mathbf{e}_v\|, \mathbf{r}_k \rangle$ of the unit-normalized embedding of each token v onto a key-seeded direction $\mathbf{r}_k$, and cuts the sorted list into q equal-count buckets. Synonyms tend to have nearby projections, so this coloring is designed to let a substitution keep the color more often than chance and thereby preserve part of the signal under editing. Unlike the semantic-cluster coloring, it keeps the false-positive rate close to the nominal value (Section E.1).

Balance holds exactly for token counts, but text does not use every token equally often. In some topics, one class then carries more or less than $1/q$ of the probability, the chance rate drifts away from $1/q$, and token frequencies begin to reveal the key (Proposition 5). ZCA whitening (Kessy et al., 2018) of the embeddings reduces this drift but does not remove it (Section C). Our defaults do not whiten, and we set the detection threshold per model on unwatermarked text.

3.2 Generation: enforcing the checksum inside the denoising loop

The red–green list (Zhao et al., 2024) favors one token set for the entire generation. TANGO instead chooses the favored class at each position from a token that is already unmasked. We say that position i is *eligible* at a denoising step if it is still masked and its tap $i - \delta$ is already unmasked. A position never waits for its tap, so generation proceeds in whatever order the sampler chooses. At an eligible position, TANGO favors the tokens that satisfy the checksum. With $a_\delta \in \mathbb{Z}_q \setminus \{0\}$ the coefficient of the tap and $b \in \mathbb{Z}_q$ the secret target residue, the checksum is

$$\big(\chi(t_i) + a_\delta\, \chi(t_{i-\delta})\big) \bmod q \;=\; b. \tag{1}$$

The new token's color enters with unit weight, so once the tap is unmasked, exactly one class satisfies Equation (1),

$$s_i \;=\; \big(b - a_\delta\, \chi(t_{i-\delta})\big) \bmod q. \tag{2}$$

TANGO adds β to the logits of every token in class s_i (after classifier-free guidance on LLaDA), and the step then samples and unmasks candidates as usual (Algorithm 1 in Section B). The general form uses a set $\mathcal{D}$ of tap lags and replaces the tap term of Equation (1) by $\sum_{\delta \in \mathcal{D}} a_\delta\, \chi(t_{i-\delta})$, so that a checksum reads $|\mathcal{D}| + 1$ tokens. In this paper we use a single tap, $\mathcal{D} = \{2\}$ (Section 4.4).

When the tap is still masked. The sampler often commits a position before its tap. With enforcement from the left only (*one-sided* enforcement), this leaves about 40% of scored pairs on LLaDA unenforced (Table 19). With a single tap and $\gcd(a_\delta, q) = 1$, Equation (1) can also be solved for the tap, so a position whose right neighbor at $i + \delta$ is already committed is biased toward the color that completes that pair. With this *either-side* enforcement, most pairs are enforced at whichever endpoint the sampler commits second. Detection and the guarantees of Section 3.4 are unchanged, because each tilted token still completes exactly one pair through the same checksum.

3.3 Detection

The detector uses only the candidate text and the key, so it meets the prompt-free requirement (R1). Given a text $t_1, \ldots, t_L$, position i is scored whenever its tap $t_{i-\delta}$ exists, that is, whenever $i > \delta$. Let $\mathcal{T}$ be the set of scored positions and $T = |\mathcal{T}|$ their number. Each scored position i indexes the pair $(t_{i-\delta}, t_i)$, so we also call T the number of scored pairs. Both depend only on the observed text, so the detector never needs the order in which positions were unmasked (R2). At each scored position the detector recolors the token and its tap with χ and checks whether Equation (1) holds. The statistic C counts the positions where it does, and z measures how far C exceeds its chance value T/q, in standard deviations of the binomial null,

$$C \;=\; \sum_{i \in \mathcal{T}} \mathbf{1}\big[\big(\chi(t_i) + a_\delta\, \chi(t_{i-\delta})\big) \bmod q = b\big], \qquad z \;=\; \frac{C - T/q}{\sqrt{T}\,\sigma_0}, \qquad \sigma_0^2 \;=\; \tfrac{1}{q}\big(1 - \tfrac{1}{q}\big), \tag{3}$$

where $\mathbf{1}[\cdot]$ is the indicator and σ_0^2 is the variance of one match under the null. The text is declared watermarked when z exceeds a threshold $z_\star$. Choosing $z_\star$ so that unwatermarked text is flagged

at a chosen false-positive rate is what we call *calibration*. Under three approximations stated in Section A.5 and with one-sided enforcement, this count test is the likelihood-ratio test among texts with the same T. At every false-positive rate that a threshold on C attains, no detector then detects more (Proposition 4).

The binomial null is exact under two conditions. First, the scored token's color is uniform over the q classes and independent of the tap's color. Adding the tap's term then gives a uniform residue, so the checksum holds with probability $1/q$ at every position. Second, the match events at different positions are independent. Then C is exactly binomial.

Proposition 1 (Binomial null). *Assume that at each scored position the color $\chi(t_i)$ is uniform on $\mathbb{Z}_q$ and independent of the colors of its taps, and that the match indicators are mutually independent across positions. Then $C \sim \mathrm{Binom}(T, 1/q)$ for every residue b, and z converges in distribution to $\mathcal{N}(0,1)$ as $T \to \infty$. (Proof in Section A.2.)*

Real text satisfies neither condition exactly. For a hash coloring, uniformity holds in expectation over the key to within $1/V$ (Proposition 3). For the semantic-quantile coloring, which is not uniformly random, we measure the class masses directly (Table 6). Independence fails because neighboring tokens are correlated, most strongly when adjacent. Under the default key, the null mean score of natural text ranges from -0.93 to $+0.98$ across residues at lag 1 and from -0.42 to $+0.24$ at lag 2 (Table 6). This correlation is why the default tap skips the adjacent token.

3.4 Theoretical guarantees

We analyze how much text the detector needs (Proposition 2), what token frequencies reveal about the key (Theorem 1), and how edits weaken the score (Theorem 2).

Setting and assumptions. Let $M_i \in \{0,1\}$ indicate whether the checksum holds at scored position i, so that $C = \sum_{i \in \mathcal{T}} M_i$. On the generation side, let p_i be the model's distribution $p_\theta(x_i = \cdot \mid x^{(j-1)})$ at position i after temperature scaling, and $m_s(p) = \sum_{v:\chi(v)=s} p(v)$ the total probability that p assigns to tokens of class s. Adding β to the logits of class s multiplies the probability of each of its tokens by e^β and renormalizes. We call the result the *tilt* of p toward s,

$$(\mathsf{T}_s p)(v) \;=\; \frac{p(v)\, e^{\beta \mathbf{1}[\chi(v)=s]}}{1 + \gamma\, m_s(p)}, \qquad \gamma \;=\; e^\beta - 1. \tag{4}$$

A token is *enforced* when the sampler draws it from a tilt, and a scored pair is enforced when its token committed second was tilted toward the class that completes it. The *enforced fraction* κ is the expected fraction of scored pairs that are enforced. The results use four idealizations, stated in full in Section A.1. **(S)** Sampling: candidates are drawn independently given the state and the favored classes, from $\mathsf{T}_{s_i} p_i$ at enforced positions and from p_i elsewhere. They are committed by the unwatermarked rule, and a committed token is treated as a draw from its candidate distribution. This idealizes the confidence selection, whose confidences come from the biased logits. **(U)** Unpredictability: given the history of states and χ, the favored classes of the positions enforced at a step are independent and uniform on $\mathbb{Z}_q$. For a real key, the favored class is a function of the tap's color. (U) drops this link, which is what the detector counts, so we use (U) only for token frequencies. The detection score uses a local form (U′) that keeps the tilt toward the pair being scored. **(B)** Balance: $m_s(p_i) = 1/q$ for every class s and position i. It holds in expectation over the key to within $1/V$ for a hash coloring (Proposition 3). **(I)** Independence: the match indicators M_i are mutually independent. It is the second hypothesis of Proposition 1. Section C measures how far unwatermarked text departs from (B) and (I).

How much text detection needs. The detector needs enough scored positions for the matches caused by the watermark to stand out from the noise of the null distribution. At an enforced position the bias raises the probability that the checksum holds above the chance value $1/q$. We call the average of this excess over all scored positions the *coupling margin* ε. The margin grows with β and shrinks when p_i is already peaked, because a confident model has little probability to move into the favored class. Under (S), (U′), and (B), unenforced pairs match at the chance rate, so $\varepsilon = \kappa\, \bar{\varepsilon}_{\mathrm{enf}}$ with $\bar{\varepsilon}_{\mathrm{enf}}$ the margin at enforced pairs (Lemma 1), and either-side enforcement raises ε by raising κ. The expected score is the ratio of the $T\varepsilon$ matches beyond chance to the null standard deviation $\sqrt{T}\,\sigma_0$.

Proposition 2 (Expected detection score). *For a fixed number T of scored positions, let $\mu_i = \Pr(M_i = 1)$ under the watermarked process and $\varepsilon = \frac{1}{T}\sum_{i \in \mathcal{T}} (\mu_i - 1/q)$. Then $\mathbb{E}[z] = \sqrt{T}\,\varepsilon/\sigma_0$.*

The identity follows from linearity of expectation and needs no independence (Section A.3). The score thus grows like the square root of the text length. With a fixed positive margin it reaches $z_\star$ once $T \geq z_\star^2 \sigma_0^2 / \varepsilon^2$ (Corollary 1, compared with LLaDA text in Figure 4), and under (I) the false-negative rate decays exponentially in T (Corollary 2).

First-order unforgeability. An attacker who reconstructs χ can forge, and requirement (R3) asks that token frequencies not make χ easy to infer. The attacker can rank tokens by their frequency shift $\Delta f(v) = f_{\mathrm{wm}}(v) - f_{\mathrm{base}}(v)$ between watermarked and unwatermarked generations, and the area under the ROC curve (AUC) of this ranking measures how well it separates the color classes. Under balance, every tilt has the same normalizer $1 + \gamma/q$ and each token is boosted by exactly one of the q tilts, so the tilts average to the unwatermarked distribution, $\frac{1}{q} \sum_s \mathsf{T}_s p = p$ (Lemma 2).

Theorem 1 (First-order unforgeability). *Fix a coloring χ. Under (S), (U), and (B), the generated text has the same distribution with and without the watermark. Hence $\mathbb{E}[\Delta f(v)] = 0$ for every $v \in \mathcal{V}$, and any attacker who ranks each token v by $g(\mathbb{E}[\Delta f(v)])$, for a fixed function g, separates the color classes no better than chance (AUC $\frac{1}{2}$). (Proof in Section A.6.)*

In words, when the favored class behaves like a fresh uniform draw at every position, the watermark pushes no token's expected frequency in a direction set by its color. The theorem does not say that one key produces unwatermarked text, because (U) drops the link that the detector counts. It also covers only expected shifts. The attacker of Section 4.3 ranks tokens by shifts measured on finitely many texts, so its chance-level result is an empirical finding. When (B) fails, the cancellation is inexact. The leak is proportional to the class imbalance and, for small γ, second order in γ (Proposition 5). The red–green list favors the same tokens at every position. Its favored class never varies, so nothing averages out, and ranking by frequency shift recovers the green list (Proposition 6).

Edit robustness. An edit breaks a checksum whenever it changes any of the $r = |\mathcal{D}| + 1$ tokens the checksum reads, so every added tap makes every checksum easier to break.

Theorem 2 (Edit robustness). *Suppose that an attacker corrupts each token with probability ρ, independently across tokens and of the text, that a corrupted token's color is uniform on $\mathbb{Z}_q$ and independent of everything else, that positions stay aligned, and that $\gcd(a_\delta, q) = 1$ for every $\delta \in \mathcal{D}$ (which holds automatically when q is prime). Then, exactly, $\mathbb{E}[z_{\mathrm{att}}] = (1-\rho)^{|\mathcal{D}|+1}\, \mathbb{E}[z_{\mathrm{clean}}]$, where z_{att} and z_{clean} are the scores of the attacked and the clean text.*

Random substitution thus multiplies the expected score by the probability that a whole checksum is untouched (proof in Section A.7). The gcd condition handles a tap that was edited. Such a tap contributes a_δ times a uniform residue. Multiplication by a_δ permutes the residues exactly when a_δ is a unit modulo q, and the checksum then holds with probability exactly $1/q$. Every added tap costs another factor of $(1-\rho)$, and the ablation of Section 4.4 shows that detection under edits falls as taps are added.

4 Experiments

4.1 Setup

TANGO configuration. Unless noted otherwise, TANGO uses the semantic-quantile coloring with $q = 3$ colors, a single tap at lag $\delta = 2$ ($a_2 = 1$), target residue $b = 0$, logit bias $\beta = 5$, and enforcement from either side, with the unwhitened coloring on both models. Section B gives the rule that selected these defaults.

Models and baselines. We evaluate LLaDA-8B-Instruct (Nie et al., 2025) and Dream-v0-Instruct-7B (Ye et al., 2025), which is initialized from Qwen2.5-7B (Yang et al., 2024) and shares its tokenizer. Prompts come from the C4 dataset (Raffel et al., 2020), and every method generates 128 tokens at temperature 1.0, with classifier-free guidance (CFG) at scale 2.0 on LLaDA (Section B). The baselines are the red–green list (the Unigram watermark of Zhao et al. (2024), with green fraction 0.25 and logit bias 5) and the prefix-keyed Gumbel rule (Aaronson & Kirchner, 2023), both applied along the confidence-ordered unmasking. Dream's sampler is incompatible with the Gumbel rule, so we run the Gumbel rule on LLaDA only. Each comparison uses $n = 200$ generations per method on LLaDA

Table 1: TANGO detects nearly all unedited and most edited watermarked texts at a perplexity close to the red–green list's. TPR@1%FPR under no attack (clean) and under the three edits, with the half-width of a 95% bootstrap interval as superscript. TANGO at the default setting, red–green list with green fraction 0.25 and bias 5, Gumbel rule at temperature $\tau = 1.0$. $n = 200$ per method on LLaDA and 100 on Dream, with the unwatermarked continuations of the same prompts as negatives. PPL is Qwen2.5-7B-Instruct perplexity over non-degenerate continuations (Section B).

			TPR@1%FPR				
model	method	AUROC	clean	del30	syn30	ins20	PPL
LLaDA	TANGO	0.998	$0.97^{\pm 0.02}$	$0.68^{\pm 0.07}$	$0.92^{\pm 0.04}$	$0.91^{\pm 0.04}$	7.2
LLaDA	Red–green list	0.993	$0.98^{\pm 0.02}$	$0.96^{\pm 0.02}$	$0.94^{\pm 0.03}$	$0.98^{\pm 0.02}$	7.6
LLaDA	Gumbel	0.815	$0.15^{\pm 0.05}$	$0.10^{\pm 0.04}$	$0.07^{\pm 0.04}$	$0.07^{\pm 0.04}$	4.6
LLaDA	unwatermarked	–	–	–	–	–	4.2
Dream	TANGO	1.000	$1.00^{\pm 0.00}$	$0.73^{\pm 0.08}$	$0.98^{\pm 0.03}$	$0.97^{\pm 0.03}$	7.8
Dream	Red–green list	1.000	$1.00^{\pm 0.00}$	$0.98^{\pm 0.03}$	$0.98^{\pm 0.03}$	$0.98^{\pm 0.03}$	9.1
Dream	unwatermarked	–	–	–	–	–	3.8

and $n = 100$ per method on Dream. We also run two published watermarks for masked-diffusion models on the same prompts, using their public code. They are the context-hashed green list of Gloaguen et al. (2026), which we call the DLM watermark, and dgMARK (Hong & No, 2026).

Attacks and metrics. Edits are word-level MarkLLM attacks (Pan et al., 2024): deleting 30% of the words (del30), replacing 30% of them with synonyms (syn30), and inserting 20% random words (ins20). Synonyms come from WordNet (Miller, 1995). We report the area under the detector's ROC curve (AUROC) and the true-positive rate (TPR) at a 1% false-positive rate (FPR), written TPR@1%FPR, both against unwatermarked text. Generation quality is measured as perplexity (PPL) under Qwen2.5-7B-Instruct (Yang et al., 2024) conditioned on the prompt (Section B gives this protocol and the full attack set). Section D.8 gives confidence intervals over three seeds and four keys.

4.2 Detection and quality

Table 1 compares TANGO with the red–green list and the Gumbel rule. TANGO detects 97% (LLaDA) and 100% (Dream) of unedited watermarked texts, and 68–92% and 73–98% after edits, with deletion the hardest attack. Its perplexity is 7.2 on LLaDA and 7.8 on Dream, against 7.6 and 9.1 for the red–green list. On Dream, the red–green list degenerates into repetition on 62% of its texts, against 31% for TANGO and 20% for unwatermarked text (Section D.5). Section G shows qualitative samples.

Of the three edits, deletion is the one where the red–green list keeps a clear lead (96% against 68% of deleted texts on LLaDA; Table 12 in Section D sweeps both biases). A deleted word breaks every TANGO pair that spans it, whereas the red–green list scores each token on its own. That list's robustness comes from applying the same vocabulary bias at every position, which is also what exposes its key (Section 4.3). Of the published diffusion watermarks, the DLM watermark lies on the red–green list's trade-off between detection and perplexity, and dgMARK detects fewer clean texts than TANGO at a larger perplexity cost (Table 11 in Section D). The Gumbel rule is distortion-free in expectation over its key and costs little perplexity. It detects poorly, however, with AUROC 0.815 and clean TPR@1%FPR 0.15, because the sampler commits only positions where the keyed draw rarely changes the token (Section 1). Raising the temperature restores detection at the price of quality. At temperature 1.5 it detects 83% of deleted texts, at perplexity 20.6 (Table 12).

4.3 Key recovery and forgery

Key recovery and forgery from token frequencies. Token frequencies reveal the red–green list's key but not TANGO's. With one-sided enforcement at $\beta = 8$, the total-variation distance between watermarked and unwatermarked token frequencies is 0.217 for TANGO. This is close to the 0.210 that finite samples alone produce, and below the 0.340 of the red–green list (Table 8

Table 2: Three colors and a single tap detect best under edits. LLaDA-8B, one-sided enforcement at $\beta = 8$, $n = 24$ per row, defaults in bold. The q rows use lag 1 and the tap rows use $q = 3$, so they are separate runs. PPL is GPT-2-large perplexity, and null FPR is the fraction of 200 natural C4 texts with $z > 4$. The residue ablation is in Section C.3.

			TPR@1%FPR					
axis	setting	AUROC	clean	del30	syn30	ins20	PPL	null FPR (%)
colors q	2	0.989	0.92	0.62	0.71	0.83	20.4	0.0
	3	**1.000**	**1.00**	**0.92**	**0.96**	**1.00**	**22.2**	**0.0**
	4	1.000	1.00	0.79	0.83	0.96	26.8	0.5
	8	1.000	1.00	0.83	0.83	1.00	29.9	0.5
	16	0.998	0.96	0.88	0.96	0.96	29.5	0.0
taps $\mathcal{D}$	$\mathbf{\{2\}}$	**1.000**	**1.00**	**0.92**	**0.96**	**0.96**	**25.0**	**0.0**
	$\{1, 2\}$	0.990	0.96	0.25	0.58	0.79	19.6	0.0
	$\{1, 2, 3\}$	0.986	0.83	0.25	0.25	0.54	16.7	0.0
	$\{1, 2, 3, 4\}$	0.984	0.83	0.12	0.38	0.42	18.9	0.0

in Section D). Ranking tokens by how much more often they appear in watermarked than in unwatermarked text recovers the green list, and the recovery improves as the attacker collects more texts, from AUC 0.738 at 25 texts to 0.806 at 100 (Figure 2, a separate run with TANGO at $\beta = 6$ and the whitened coloring). Against TANGO, recovery stays at chance at every number of texts (0.504 at 200 texts on LLaDA and 0.501 on Dream). An attacker can then generate text that favors the tokens of the recovered key. Every such forgery passes the red–green detector, and none passes TANGO's (Table 8).

Key recovery and forgery from token pairs. An attacker who knows TANGO's design can instead count token *pairs* at the tap lag and rank them by how much more often they appear in watermarked text. This attack is much harder, because it must estimate the frequencies of $O(V^2)$ pairs instead of $O(V)$ tokens, but the attacker may still find some signal. Scored against the true key, the ranking separates pairs that satisfy the checksum from the rest with AUC 0.62 at 200 texts, and 0.78 on the pairs seen at least three times. Turning that signal into a forgery succeeds far less often than against the red–green list. We let an attacker who has the model but not the key collect the over-represented pairs from 200 texts and bias its own generation toward them. We score forgeries at each detector's 1% false-positive threshold. At the bias level that lets 51% of forgeries pass the red–green detector, only 4% pass TANGO's (Table 10 in Section D). At this bias the forgeries stay fluent, with GPT-2-large perplexity 13.5 against 13.1 for unwatermarked text. At twice the bias, 55% pass TANGO's detector and 98% pass the red–green detector. A simpler attacker that strings over-represented pairs together without a language model shows the same ordering at every number of texts (74% against 96% at 200 texts, Table 9). Increasing the number of taps may further weaken this attacker, at the cost of lower detection, especially under edits (Section 4.4).

4.4 Ablations

Table 2 varies TANGO's two main design choices, the number of colors q and the tap set $\mathcal{D}$, on LLaDA. The target residue $b = 0$ is the one that unwatermarked text satisfies least often under the default key, because the colors of neighboring tokens are correlated (Section C.3).

Colors and taps. Detection under edits improves sharply from $q = 2$ to $q = 3$ (del30 TPR $0.62 \to 0.92$) and does not improve further, while GPT-2-large perplexity rises from 22.2 at $q = 3$ to 29.9 at $q = 8$, so we use $q = 3$. Consistent with Theorem 2, deletion TPR drops from 0.92 with one tap to 0.12 with four (Figure 3 in Section E), so we use a single tap.

Enforced fraction. The enforced fraction κ sets the margin of Proposition 2. With one-sided enforcement at $\beta = 8$, κ is 0.61 on LLaDA. Either-side enforcement raises it to 0.74 at the same bias, and to 0.76 at the default $\beta = 5$ (Table 19 in Section E). Because $\varepsilon = \kappa\, \bar{\varepsilon}_{\mathrm{enf}}$, raising κ from 0.61 to 0.74 at a fixed margin per enforced pair raises the expected score by a factor of about 1.2.

5 Related Work

Biased and distortion-free watermarks. Statistical text watermarks differ in whether they change token frequencies. Biased watermarks do. A keyed green list adds a logit bias to a vocabulary subset that is either derived from a hash of the context (Kirchenbauer et al., 2023) or fixed once (Zhao et al., 2024). The resulting frequency shift lets an attacker approximate the green list and forge or remove the watermark (Jovanović et al., 2024; Zhang et al., 2024b). Reweighting watermarks change the next-token distribution under a fixed key but preserve it in expectation over keys (Hu et al., 2024; Wu et al., 2024), and distortion-free watermarks use the key to seed the randomness of sampling (Aaronson & Kirchner, 2023; Kuditipudi et al., 2024; Christ et al., 2024; Dathathri et al., 2024). Like biased watermarks, TANGO biases the sampler, so detection counts matches and needs no prompt. Unlike them, it favors a different class at each position. Under the idealized sampler of Theorem 1, expected token frequencies are then unchanged, and the signal appears in pair statistics.

Watermarks for diffusion language models. The DLM watermark of Gloaguen et al. (2026) applies a context-hashed red–green bias in expectation over still-masked context, and dgMARK (Hong & No, 2026) steers the unmasking order so that the final text carries a statistical signature. Other diffusion watermarks add predicted and bidirectional context to the context hash (Wu et al., 2025), apply Gumbel-max sampling at every denoising step with independent (Bagchi et al., 2025) or locally correlated (Li & He, 2026) perturbations, or control a global sketch of the whole sequence (Zhao, 2026). The closest work, LR-DWM (Raban et al., 2026), hashes the token id of the left and of the right neighbor, whenever each is unmasked, into two green lists and biases the token toward both. TANGO also keys on an unmasked neighbor, but only through its color, so the favored set is always one of q balanced classes, and a substitution that keeps the neighbor's color keeps the favored class. Section F discusses these works in more detail, along with semantic watermarks, watermark stealing, and signatures.

6 Discussion

Most text watermarks assume left-to-right generation, while masked-diffusion language models generate tokens in no fixed order. The red–green list still runs in this setting, but it exposes its key through token frequencies. TANGO embeds its signal in pairs of tokens during the denoising loop. Under our idealized model, the expected frequency of every token is then unchanged (Theorem 1). In our experiments, key recovery from token frequencies stays at chance, and forgeries built from those frequencies fail. Detection needs only the candidate text and the key.

Scope of the guarantees. Theorem 1 covers expected token frequencies under an idealized sampler. An attacker who counts token pairs can forge TANGO, though less often than the red–green list in every setting we test. In a word-level test, even an attacker who holds the key and uses it to choose which words to regenerate removes the watermark about as fast as random editing does, or slower (Table 14 in Section D). Rotating the key per deployment window limits how many texts an attacker can collect under one key, at the cost of testing each candidate text against every active key. Key rotation is the mitigation we recommend in practice.

Robustness and calibration. TANGO trades some robustness to deletion for a hidden key. The red–green list survives deletion better at matched perplexity because it biases the same tokens at every position, which is the property that exposes its key. Theorem 2 covers only edits that keep positions aligned. Paraphrase by back-translation moves most taps and lowers TANGO's detection more than the red–green list's (Section D.7). Dream's natural text scores above chance, so the detection threshold should be set on unwatermarked text for each new model (Section C).

Which watermark to deploy. A provider whose detections must hold up against forgery gains the most from TANGO. Under synonym substitution and word insertion, TANGO detects at most seven percentage points fewer watermarked texts than the red–green list on both models (Table 1). Its key could not be read from token frequencies in our experiments, and every attack we test forges it less often than the red–green list. For such a provider, we argue that a hidden key is worth some loss of robustness to deletion.

AI Use Statement

The authors developed the research questions and core conceptual framework and directed the methodology and interpretation. Generative AI assisted with proof writing, literature review, code implementation, figure preparation, and prose polishing. The authors take responsibility for the final content.

Reproducibility Statement

Section 3 and Algorithm 1 specify how TANGO generates and detects the watermark. Assumptions accompany the theoretical results in Section 3.4 and Section A.1, and Section A contains the complete proofs. Section 4 summarizes the experiments, with prompts, decoding settings, perplexity protocols, edit attacks, hyperparameter selection, and software versions in Section B, attack configurations and sample sizes in Section D, and variation over seeds and keys in Section D.8. Code, configurations, generated texts, and results are at `https://anonymous.4open.science/r/tango-watermark`.

References

Scott Aaronson and Hendrik Kirchner. Watermarking of large language models. Talk, Simons Institute for the Theory of Computing, 2023.

Kasra Arabi, Benjamin Feuer, R. Teal Witter, Chinmay Hegde, and Niv Cohen. Hidden in the noise: Two-stage robust watermarking for images. In *International Conference on Learning Representations (ICLR)*, 2025a.

Kasra Arabi, R. Teal Witter, Chinmay Hegde, and Niv Cohen. Seal: Semantic aware image watermarking. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b. doi: 10.1109/ICCV51701.2025.01503.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Avi Bagchi, Akhil Bhimaraju, Moulik Choraria, Daniel Alabi, and Lav R. Varshney. Watermarking discrete diffusion language models. *arXiv preprint arXiv:2511.02083*, 2025.

Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. De-mark: Watermark removal in large language models. *arXiv preprint arXiv:2410.13808*, 2024.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *Conference on Learning Theory (COLT)*, volume 247 of *PMLR*, pp. 1125–1139, 2024.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/biomet/26.4.404.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634:818–823, 2024. doi: 10.1038/s41586-024-08025-4.

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. Publicly-detectable watermarking for language models. *IACR Communications in Cryptology*, 1(4), 2025. doi: 10.62056/ahmpdkp10.

Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. Discovering spoofing attempts on language model watermarks. *arXiv preprint arXiv:2410.02693*, 2024.

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, and Martin Vechev. Watermarking diffusion language models. In *International Conference on Learning Representations (ICLR)*, 2026.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. On the learnability of watermarks for language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Pyo Min Hong and Albert No. dgMARK: Decoding-guided watermarking for diffusion language models. In *International Conference on Machine Learning (ICML)*, 2026.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. SemStamp: A semantic watermark with paraphrastic robustness for text generation. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Nikola Jovanović, Robin Staab, and Martin Vechev. Watermark stealing in large language models. In *International Conference on Machine Learning (ICML)*, 2024.

Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning (ICML)*, 2023.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research (TMLR)*, 2024.

Baixin Li and Haiyun He. SAC-Copula: Quality-preserving watermarking for diffusion language models via smooth correlated Gumbel fields. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026.

Huijia Lin, Kameron Shahabi, and Min Jae Song. Unforgeable watermarks for language models via robust signatures. IACR ePrint 2026/282 (also arXiv:2602.15323), 2026.

Aiwei Liu, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S. Yu. An unforgeable publicly verifiable watermark for large language models. In *International Conference on Learning Representations (ICLR)*, 2024a.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. In *International Conference on Learning Representations (ICLR)*, 2024b.

Tianhao Ma, Weihao Xuan, Dong-Dong Wu, Farshid Nooshi, Takashi Ishida, Gang Niu, Naoto Yokoya, and Masashi Sugiyama. DenMark: Robust semantic watermarking for diffusion language models. *arXiv preprint arXiv:2609.14257*, 2026.

George A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11): 39–41, 1995. doi: 10.1145/219717.219748.

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.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. MarkLLM: An open-source toolkit for LLM watermarking. In *Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, 2024.

Ofek Raban, Gal Chechik, and Ethan Fetaya. LR-DWM: Efficient watermarking for diffusion language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 43510–43517, 2026. doi: 10.18653/v1/2026.findings-acl.2161.

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

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. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. A robust semantics-based watermark for large language model against paraphrasing. In *Findings of the Association for Computational Linguistics: NAACL*, pp. 613–625, 2024.

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. *Transactions on Machine Learning Research (TMLR)*, 2025.

Subham Sekhar 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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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. *arXiv preprint arXiv:2510.02902*, 2025.

Qilong Wu and Varun Chandrasekaran. Bypassing LLM watermarks with color-aware substitutions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 8549–8581, 2024. doi: 10.18653/v1/2024.acl-long.464.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. A resilient and accessible distribution-preserving watermark for large language models. In *International Conference on Machine Learning (ICML)*, 2024.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7B: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for generative models. In *International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, 2024a.

Zhaoxi Zhang, Xiaomei Zhang, Yanjun Zhang, Leo Yu Zhang, Chao Chen, Shengshan Hu, Asif Gill, and Shirui Pan. Large language model watermark stealing with mixed integer programming. *arXiv preprint arXiv:2405.19677*, 2024b.

Daniel Zhao. Global sketch-based watermarking for diffusion language models. *arXiv preprint arXiv:2606.04486*, 2026.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for AI-generated text. In *International Conference on Learning Representations (ICLR)*, 2024.

Xuandong Zhao, Sam Gunn, Miranda Christ, Jaiden Fairoze, Andres Fabrega, Nicholas Carlini, Sanjam Garg, Sanghyun Hong, Milad Nasr, Florian Tramer, Somesh Jha, Lei Li, Yu-Xiang Wang, and Dawn Song. SoK: Watermarking for AI-generated content. In *IEEE Symposium on Security and Privacy (S&P)*, 2025.

Tong Zhou, Xuandong Zhao, Xiaolin Xu, and Shaolei Ren. Bileve: Securing text provenance in large language models against spoofing with bi-level signature. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Appendix Contents

A Proofs 15

A.1 Setting, notation, and assumptions 15

A.2 The score of unwatermarked text 18

A.3 Expected score and the text length needed for detection 19

A.4 Finite-sample error bounds 21

A.5 The count test is the likelihood-ratio test 22

A.6 Token frequencies and the key 23

A.7 Edit robustness 26

B Experimental details 27

C Calibration of the detector on unwatermarked text 29

C.1 LLaDA 29

C.2 Dream 29

C.3 Correlated colors and the choice of residue 29

D Full results 30

D.1 Key recovery and forgery from token frequencies 30

D.2 Key recovery and forgery from token pairs 31

D.3 The published diffusion watermarks 32

D.4 Detection against perplexity at every strength 33

D.5 Results on Dream 34

D.6 Removal by an attacker who holds the key 34

D.7 Paraphrase 35

D.8 Variance over seeds and keys 35

E Full ablations 35

E.1 Number of colors 36

E.2 Tap set 36

E.3 Enforcement side, bias, lag, and unmasking order 37

E.4 Detection grows with text length 38

F Additional related work 38

G Qualitative samples 39

Section A proves the formal results of Section 3 and states every assumption they use. Section B gives the decoding and evaluation settings and the rule that chose TANGO's defaults. Section C checks that the detection threshold controls false positives on unwatermarked text from both models and explains where such text departs from the null. Section D gives the full tables behind Section 4, including the Dream results, the published diffusion watermarks, the attacks that the main text only mentions, and checks over paraphrase, seeds, and keys. Section E gives the full ablations. Section F covers further related work, and Section G shows generated text.

A Proofs

This appendix proves the four formal statements of Section 3 and eight supporting results. Table 3 lists every result, what it says, and the assumptions it uses. Section A.1 first restates the notation and the assumptions, so that the appendix can be read without returning to the main text.

Table 3: The formal results, what each says, and what each assumes. The assumptions (S), (U), (U′), (B), and (I) are stated in Section A.1. Results marked † also treat a committed token as a draw from its candidate distribution, the second part of (S).

result	claim in words	assumes	appendix
Proposition 1	Unwatermarked text gives a $\mathrm{Binom}(T, 1/q)$ count.	uniform colors, (I)	A.2
Proposition 3	A uniformly random balanced coloring gives every class mass close to $1/q$ unless the context is peaked.	hash coloring	A.2
Proposition 2	The expected score is $\sqrt{T}\,\varepsilon/\sigma_0$.	definitions only, fixed T	A.3
Corollary 1	Detection needs $T_\star = z_\star^2\sigma_0^2/\varepsilon^2$ scored positions.	$z_\star > 0$, constant $\varepsilon > 0$	A.3
Lemma 1†	Unenforced pairs match at the chance rate, so $\varepsilon = \kappa\,\bar{\varepsilon}_{\mathrm{enf}}$.	(S), (U′), (B), single tap	A.3
Corollary 2	The false-negative rate decays exponentially in T.	(I)	A.4
Proposition 4†	With one-sided enforcement, the count test is the likelihood-ratio test.	Approximations 1–3	A.5
Lemma 2	Averaging the q tilts returns the context.	(B)	A.6
Theorem 1	For a fixed coloring, expected token frequencies do not change.	(S), (U), (B)	A.6
Proposition 5	Without balance, the frequency shift is bounded by the imbalance.	fixed context	A.6
Proposition 6	The red–green list shifts frequencies in the direction of its key.	fixed contexts	A.6
Theorem 2	Random substitutions scale the expected score by $(1-\rho)^{\lvert\mathcal{D}\rvert+1}$.	random substitutions, $\gcd(a_\delta, q) = 1$	A.7

A.1 Setting, notation, and assumptions

This subsection restates the checksum, the tilt, and the four assumptions that the proofs below use. Table 4 collects the symbols.

The checksum. A secret key fixes a balanced coloring $\chi : \mathcal{V} \to \mathbb{Z}_q$ of the vocabulary into q classes and a target residue $b \in \mathbb{Z}_q$. Here $\mathbb{Z}_q = \{0, \dots, q-1\}$ with addition and multiplication modulo q, and *balanced* means that every class has $\lfloor V/q \rfloor$ or $\lceil V/q \rceil$ tokens. The edit-robustness result and the tap ablation of Section E use several taps, so the proofs use a general checksum with a set $\mathcal{D}$ of tap lags and coefficients $a_\delta \in \mathbb{Z}_q \setminus \{0\}$. The main text's single tap is the case $\mathcal{D} = \{2\}$ with $a_2 = 1$. In a text $t_1, \dots, t_L$, position i is scored when all its taps exist, and $\mathcal{T}$ is the set of scored positions, $T = |\mathcal{T}|$. For a scored position i, the residue and the match indicator are

$$R_i \;=\; \Big(\chi(t_i) + \sum_{\delta\in\mathcal{D}} a_\delta\,\chi(t_{i-\delta})\Big) \bmod q, \qquad M_i \;=\; \mathbf{1}[R_i = b],$$

where $\mathbf{1}[\cdot]$ is the indicator. The detector computes

$$C \;=\; \sum_{i\in\mathcal{T}} M_i, \qquad z \;=\; \frac{C - T/q}{\sqrt{T}\,\sigma_0}, \qquad \sigma_0^2 \;=\; \tfrac{1}{q}\big(1-\tfrac{1}{q}\big),$$

as in Equation (3), and flags the text when $z > z_\star$. Here σ_0^2 is the variance of one match under the null.

The tilt. Let p_i be the model's distribution over the vocabulary at position i after temperature scaling. For brevity, in these proofs we call such a distribution a *context*, since it is what the model predicts from the tokens around position i. Let $m_s(p) = \sum_{v:\chi(v)=s} p(v)$ be the total probability that a distribution p assigns to tokens of class s. We call $m_s(p)$ the *class mass* of s. Adding $\beta > 0$

Table 4: Notation used in the proofs.

symbol	meaning
$\mathcal{V}$, V	vocabulary and its size, $V = \lvert\mathcal{V}\rvert$
$t_1, \dots, t_L$	tokens of a text of length L
$x^{(j)}$	state after j denoising steps, the partly unmasked sequence
q, $\mathbb{Z}_q$	number of colors, and $\{0, \dots, q-1\}$ with arithmetic modulo q
χ	balanced coloring $\mathcal{V} \to \mathbb{Z}_q$ fixed by the key; class s has $n_s \in \{\lfloor V/q \rfloor, \lceil V/q \rceil\}$ tokens
$\mathcal{D}$, δ, a_δ	set of tap lags, one lag, and its coefficient in $\mathbb{Z}_q \setminus \{0\}$
b	target residue in $\mathbb{Z}_q$, fixed by the key
R_i, M_i	residue and match indicator at scored position i
$\mathcal{T}$, T	set of scored positions and its size (the number of scored pairs)
C, z, $z_\star$	match count, detection score, and detection threshold
σ_0^2	$\frac{1}{q}(1-\frac{1}{q})$, the variance of one match under the null
β, γ	logit bias and $\gamma = e^\beta - 1$
p_i, $m_s(p)$	context (model distribution) at position i, and the class mass, the total probability p assigns to tokens of class s
$\mathsf{T}_s p$	tilt of p toward class s
s_i	favored class at position i (for a real key, the class that completes its checksum, Equation (2))
κ	enforced fraction, the expected fraction of scored pairs that are enforced
μ_i, ε	match probability at position i and coupling margin
$\bar{\varepsilon}_{\mathrm{enf}}$	excess match probability at an enforced pair under balance
$\Delta f(v)$	frequency shift of token v between watermarked and unwatermarked generations
h_j, P_j, P_j^0	history $(x^{(0)}, \dots, x^{(j)})$, and its law with and without the watermark
$\bar{Z}$	$1 + \gamma/q$, the normalizer of a tilt under balance
ρ	substitution rate of the edit attack
A_i	positions whose tokens the checksum at position i reads

to the logits of class s multiplies the probability of each of its tokens by e^β and renormalizes. With $\gamma = e^\beta - 1$ the result is the tilt

$$(\mathsf{T}_s p)(v) \;=\; \frac{p(v)\, e^{\beta \mathbf{1}[\chi(v)=s]}}{1 + \gamma\, m_s(p)}.$$

Once the taps of position i are committed, exactly one class completes its checksum,

$$s_i \;=\; \Big(b - \sum_{\delta \in \mathcal{D}} a_\delta\, \chi(t_{i-\delta})\Big) \bmod q,$$

which is Equation (2) for a general tap set. The tilt favors this class, so we also call it the *favored class*. All experiments sample at temperature $\tau = 1$. Algorithm 1 adds β to the logits before dividing by τ, so for general τ the temperature-scaled distribution is tilted by $e^{\beta/\tau}$ instead of e^β.

Enforced pairs. This paragraph and Lemma 1 concern a single tap with $\gcd(a_\delta, q) = 1$, as Algorithm 1 requires, so that once one token of a pair is committed, exactly one class completes the pair. The detector counts pairs, but the sampler tilts tokens, so we need to say which pairs the tilt acts on. A position is *committed* when the sampler unmasks it, and its token is then fixed for the rest of generation. A token is *enforced* when the sampler draws it from a tilt. A scored pair is *enforced* when the token of the pair that was committed second was tilted toward the class that completes this pair. With one-sided enforcement a token is tilted only toward the pair it forms with its tap. Either-side enforcement is defined for a single tap, and there a token can instead be tilted toward the pair it forms with its right neighbor at lag δ. Each tilted token completes exactly one pair, so tilted tokens and enforced scored pairs correspond one to one. The one exception is a token whose tap lies in the prompt. Algorithm 1 tilts it, but the detector does not score its pair. There are at most $\max \mathcal{D}$ such tokens per text, and we ignore them. Whether a pair is enforced depends on the unmasking order, so it is random. The *enforced fraction* is the expected fraction of scored pairs

that are enforced,

$$\kappa \;=\; \frac{1}{T}\sum_{i\in\mathcal{T}} \Pr(\text{pair } i \text{ is enforced}),$$

and Table 19 reports its empirical value.

Assumptions. The results use four idealizations of the sampler and the text, and a local form (U′) of the second.

- **(S) Sampling.** At each denoising step the sampler draws a candidate at every masked position, independently given the current *state* (the partly unmasked sequence) and the favored classes of that step. An enforced candidate is drawn from $\mathsf{T}_{s_i} p_i$ and every other candidate from p_i. The sampler then commits candidates by the same *commit rule* as the unwatermarked sampler. In Algorithm 1 the commit rule keeps the K most confident candidates. (S) is exact for the candidate draw. For the commit rule it is an idealization, because the confidences come from the biased logits and so differ from those of the unwatermarked sampler.

  Committed tokens. Lemma 1 and Section A.5 also treat a committed token as a draw from its candidate distribution. This is exact when the commit rule ignores the candidates, and an idealization otherwise. Theorem 1 does not need it, because both processes share the candidate distributions and the commit rule.
- **(U) Unpredictability.** At each denoising step, given the history of states $x^{(0)}, \ldots, x^{(j-1)}$ and the coloring χ, the favored classes s_i of the positions enforced in that step are independent and uniform on $\mathbb{Z}_q$.
- **(U′) Local unpredictability.** Fix a scored pair i, and let c be the class that completes it once one of its tokens is committed. A token tilted toward pair i is tilted toward c, as in Algorithm 1. Given the history of states, the coloring, and c, the favored class of every token tilted toward a different pair is uniform on $\mathbb{Z}_q$, and the favored classes of distinct tokens are independent. In short, (U′) is (U) applied to every tilt except the tilt toward the pair being scored.
- **(B) Balance.** Every context spreads its mass evenly over the classes, $m_s(p_i) = 1/q$ for every s and i. Proposition 3 shows that a uniformly random balanced coloring, such as the hash coloring, achieves this balance in expectation to within $1/V$. The semantic-quantile coloring is not uniformly random, so for it (B) is an assumption, and Section C.3 measures the class masses under the default key.
- **(I) Independence.** The match indicators M_i are mutually independent across scored positions. (I) is the second hypothesis of Proposition 1. Beyond that, it enters only the finite-sample bounds of Section A.4 and Approximation 3 of Section A.5.

What (U) idealizes. For a real key the favored class s_i is a function of the coloring, the residue, and the committed taps, so (U) cannot hold exactly. (U) describes an idealized sampler in which each enforced position draws its favored class afresh, as if every pair had its own secret residue. Like the real sampler, it favors a class that changes from position to position. It also spreads the favored classes evenly over the classes, which the real sampler does when the colors of the taps are evenly spread. Because the fresh draw is independent of the coloring, conditioning on the color of any token leaves it uniform.

For a real key, (U) is therefore an approximation. Its accuracy depends on how evenly the colors of the taps are spread over the classes and how unrelated they are to the color of the token drawn at the tilted position. Natural text meets neither condition exactly (Table 6). The red–green list favors the same class at every position. Its favored class never varies, so no sampler of the kind (U) describes approximates it.

Why detection uses (U′). (U) also drops the link between a pair's favored class and the tap of that pair, and this link is what the detector counts. Under (S), (U), and (B) the generated text has the unwatermarked law (Theorem 1), so every scored pair matches with probability $1/q$ and the margin ε is zero. The reason is that (U) turns the tilt toward a pair into a tilt toward a uniformly drawn class. With probability $(q-1)/q$ this class is not the one that completes the pair, and the pair then

matches with probability $1/(q+\gamma)$, below chance. This deficit cancels the excess of Equation (5) exactly. (U) and (U′) idealize the same sampler for two different observers. An attacker who counts single tokens never sees the tap that a token was tilted toward. What matters to this attacker is whether the favored class at a position is evenly spread and unrelated to the color of the token drawn there, which is the property that (U) idealizes. The detector reads each token together with its tap, so an analysis of its score must keep the tilt toward the pair being scored. (U′) keeps it and randomizes only the tilts toward other pairs. We therefore use (U) for Theorem 1 and (U′) for Lemma 1. Neither describes a real key exactly.

A.2 The score of unwatermarked text

We call the distribution of the score z on text written without the watermark the *null*. The threshold $z_\star$ controls the false-positive rate only if such text satisfies the checksum at rate $1/q$. Proposition 1 gives conditions under which the count is exactly binomial, and Proposition 3 shows how close a uniformly random balanced coloring comes to meeting them.

Proposition 1 (restated). *Assume that at each scored position the color $\chi(t_i)$ is uniform on $\mathbb{Z}_q$ and independent of the colors of its taps, and that the match indicators are mutually independent across positions. Then $C \sim \mathrm{Binom}(T, 1/q)$ for every residue b, and z converges in distribution to $\mathcal{N}(0,1)$ as $T \to \infty$.*

Proof of Proposition 1. The proof uses only the two hypotheses and the fact that the scored token's color enters R_i with coefficient 1.

Step 1: each position matches with probability $1/q$. Condition on the tap colors of position i. They fix the tap contribution

$$y \;=\; \sum_{\delta\in\mathcal{D}} a_\delta\, \chi(t_{i-\delta}) \bmod q, \qquad \text{so that} \qquad R_i \;=\; \big(\chi(t_i)+y\big) \bmod q.$$

The map $x \mapsto (x+y) \bmod q$ is a bijection of $\mathbb{Z}_q$ for every y. By hypothesis $\chi(t_i)$ is uniform and independent of the tap colors, so it stays uniform after conditioning, and a bijection maps a uniform variable to a uniform variable. Hence R_i is uniform given the taps, and

$$\Pr\big(M_i = 1 \mid \text{taps}\big) \;=\; \Pr\big(R_i = b \mid \text{taps}\big) \;=\; \tfrac{1}{q}$$

for every residue b. The right-hand side does not depend on the taps, so $\Pr(M_i = 1) = 1/q$.

Step 2: the count is binomial and the score is asymptotically normal. The M_i are independent by hypothesis, so $C \sim \mathrm{Binom}(T, 1/q)$, with mean T/q and variance $T\sigma_0^2$. The score z is C standardized by this mean and variance, and the de Moivre–Laplace theorem gives $z \Rightarrow \mathcal{N}(0,1)$ as $T \to \infty$ at fixed q. □

The proof does not use the condition $\gcd(a_\delta, q) = 1$. That condition lets Algorithm 1 solve Equation (1) for the left token of a pair, and it handles a corrupted tap in Theorem 2.

Where balance enters. The hypotheses of Proposition 1 do not mention balance. Balance is what makes the uniformity hypothesis plausible, because it leaves no color class larger than another. A token drawn uniformly from the vocabulary lands in class s with probability n_s/V, where $n_s \in \{\lfloor V/q \rfloor, \lceil V/q \rceil\}$ is the class size, and $|n_s/V - 1/q| < 1/V$. Real contexts are far from uniform over the vocabulary, so the relevant question is how much probability a given context p assigns to each class. The next proposition answers it for a coloring drawn uniformly from the balanced colorings, which is how the hash coloring of Section 3.1 is built.

Proposition 3 (A random key nearly balances every context). *Let χ be drawn uniformly from the balanced colorings with class sizes $n_0, \dots, n_{q-1}$, and fix a distribution p on $\mathcal{V}$. Write $\|p\|_2^2 = \sum_v p(v)^2$ for the collision probability of p, which is $1/V$ for the uniform distribution and 1 for a point mass. Then for every class s,*

$$\Big|\mathbb{E}_\chi[m_s(p)] - \tfrac{1}{q}\Big| \;<\; \frac{1}{V}, \qquad \mathrm{Var}_\chi[m_s(p)] \;=\; \frac{n_s(V-n_s)}{V(V-1)}\Big(\|p\|_2^2 - \frac{1}{V}\Big),$$

and by Chebyshev's inequality, for every $\zeta > 0$,

$$\Pr_{\chi}\Big(\big|m_s(p) - \mathbb{E}_{\chi}[m_s(p)]\big| \geq \zeta\Big) \;\leq\; \frac{\mathrm{Var}_{\chi}[m_s(p)]}{\zeta^2}.$$

Proof. Under a uniformly random balanced coloring with fixed class sizes, the set $\chi^{-1}(s)$ is a uniformly random subset of $\mathcal{V}$ of size n_s. Let $J_v = \mathbf{1}[\chi(v) = s]$, so that $m_s(p) = \sum_v p(v)\, J_v$.

Step 1: the mean. Each J_v has mean n_s/V, so $\mathbb{E}_{\chi}[m_s(p)] = n_s/V$, which is within $1/V$ of $1/q$.

Step 2: the variance. Each indicator has $\mathrm{Var}(J_v) = \frac{n_s}{V}(1 - \frac{n_s}{V})$, and for $u \neq v$

$$\mathrm{Cov}(J_u, J_v) \;=\; \frac{n_s(n_s-1)}{V(V-1)} - \frac{n_s^2}{V^2} \;=\; -\frac{1}{V-1}\cdot\frac{n_s}{V}\Big(1 - \frac{n_s}{V}\Big).$$

The variance of the weighted sum is

$$\mathrm{Var}_{\chi}[m_s(p)] \;=\; \sum_v p(v)^2\,\mathrm{Var}(J_v) + \sum_{u\neq v} p(u)\,p(v)\,\mathrm{Cov}(J_u, J_v).$$

Substituting the two moments and using $\sum_{u\neq v} p(u)p(v) = 1 - \|p\|_2^2$,

$$\mathrm{Var}_{\chi}[m_s(p)] \;=\; \frac{n_s}{V}\Big(1 - \frac{n_s}{V}\Big)\Big[\|p\|_2^2 - \frac{1 - \|p\|_2^2}{V-1}\Big] \;=\; \frac{n_s(V-n_s)}{V(V-1)}\Big(\|p\|_2^2 - \frac{1}{V}\Big),$$

where the second equality uses $\|p\|_2^2 - \frac{1-\|p\|_2^2}{V-1} = \frac{V}{V-1}\big(\|p\|_2^2 - \frac{1}{V}\big)$.

Step 3: the tail. Chebyshev's inequality gives the tail bound. □

For $n_s \approx V/q$ the variance is close to $\sigma_0^2\,\|p\|_2^2$. A random key therefore balances a spread-out context well and a peaked context poorly. At $q = 3$, a context with collision probability 0.01 has class masses with standard deviation about 0.05 across keys, and a context that puts almost all its mass on one token has class masses near 0 or 1 for every key.

The proposition does not cover the semantic-quantile coloring. Its classes are rank buckets of a keyed projection, so a token whose embedding projects near the median for most directions lands in the middle class for most keys. For this coloring $\mathbb{E}_{\chi}[m_s(p)]$ can therefore differ from $1/q$ by more than $1/V$.

One fixed key. Proposition 3 averages over keys, but a deployment uses one key, and its class masses on natural text can be unequal. Under the default key the three classes carry 32%, 29%, and 39% of the tokens of natural text (Table 6). This imbalance is why we calibrate the threshold on unwatermarked text (Section C).

A.3 Expected score and the text length needed for detection

Proposition 2 relates the expected score to the coupling margin ε, which is how far the match probability exceeds the chance rate $1/q$, averaged over the T scored positions. The proposition needs no assumption, because it uses only linearity of expectation. Texts can end early, so T varies across texts. The statements of this subsection then hold conditionally on T.

Proposition 2 (restated). *For a fixed number T of scored positions, let $\mu_i = \Pr(M_i = 1)$ under the watermarked process and $\varepsilon = \frac{1}{T}\sum_{i\in\mathcal{T}}(\mu_i - 1/q)$. Then $\mathbb{E}[z] = \sqrt{T}\,\varepsilon/\sigma_0$.*

Proof of Proposition 2. Linearity of expectation gives $\mathbb{E}[C] = \sum_i \mu_i = T(1/q + \varepsilon)$. The score z is an affine function of C, so

$$\mathbb{E}[z] \;=\; \frac{\mathbb{E}[C] - T/q}{\sqrt{T}\,\sigma_0} \;=\; \frac{T\varepsilon}{\sqrt{T}\,\sigma_0} \;=\; \sqrt{T}\,\frac{\varepsilon}{\sigma_0}.$$ □

Setting the expected score equal to the threshold gives the number of scored positions that detection needs.

Corollary 1 (Scored positions to detect). *If $z_\star > 0$ and the margin $\varepsilon > 0$ does not depend on T, then $\mathbb{E}[z] \geq z_\star$ if and only if*

$$T \;\geq\; T_\star \;=\; \frac{z_\star^2\,\sigma_0^2}{\varepsilon^2}.$$

Proof. By Proposition 2, $\mathbb{E}[z] = \sqrt{T}\,\varepsilon/\sigma_0$ increases in T when $\varepsilon > 0$ and equals $z_\star$ at $T = T_\star$. □

The number of scored positions that detection needs grows as $1/\varepsilon^2$, so halving the margin quadruples it.

The margin at an enforced pair. The margin depends on how much probability the tilt moves into the favored class. Consider an enforced pair whose second token has context p and favored class s, and let $m = m_s(p)$. If we treat the committed token as a draw from $\mathsf{T}_s p$, the pair matches exactly when that token lands in class s, which has probability

$$m_s(\mathsf{T}_s p) \;=\; \frac{(1+\gamma)\,m}{1+\gamma m}.$$

This exceeds m for every $0 < m < 1$ and equals m at $m \in \{0, 1\}$. A confident context, whose mass sits almost entirely inside or outside the favored class, therefore gains almost nothing from the tilt. Under (B) we have $m = 1/q$, and the excess over chance at an enforced pair is

$$\bar{\varepsilon}_{\mathrm{enf}} \;=\; \frac{(1+\gamma)/q}{1+\gamma/q} - \frac{1}{q} \;=\; \frac{\gamma\,(1-1/q)}{q+\gamma}, \tag{5}$$

which increases in β and saturates at $1 - 1/q$. Raising β therefore increases the margin with diminishing returns.

The main text factors the margin as $\varepsilon = \kappa\,\bar{\varepsilon}_{\mathrm{enf}}$. This needs one more fact, that a pair the sampler did not enforce matches only at the chance rate. The lemma uses the local assumption (U′), because under (U) itself every pair matches at the chance rate and the margin is zero (Section A.1).

Lemma 1 (Margin factorization). *Under (S), (U′), and (B), for a single tap with $\gcd(a_\delta, q) = 1$, an enforced pair matches with probability $1/q + \bar{\varepsilon}_{\mathrm{enf}}$, with $\bar{\varepsilon}_{\mathrm{enf}}$ as in Equation* (5)*, and an unenforced pair matches with probability $1/q$. Hence $\varepsilon = \kappa\,\bar{\varepsilon}_{\mathrm{enf}}$.*

Proof. Fix a scored pair i and apply (U′) to it. Condition on the history of states and on the coloring, and treat each committed token as a draw from its candidate distribution (second part of (S)). When the two tokens of the pair are committed at different steps, the token committed second decides whether the pair matches, because the other token is already fixed when it is drawn. Let c be the class that completes the pair given that other token. The history and the coloring fix c.

Step 1: an enforced pair matches with probability $1/q + \bar{\varepsilon}_{\mathrm{enf}}$. The second token is tilted toward c, the tilt that (U′) keeps as in Algorithm 1. It lands in c with probability $m_c(\mathsf{T}_c p)$, which is $1/q + \bar{\varepsilon}_{\mathrm{enf}}$ by the computation above with $m = 1/q$ from (B). This holds given every history in which the pair is enforced.

Step 2: an unenforced pair matches with probability $1/q$. A token tilted toward pair i is tilted toward c, so the pair is unenforced exactly when its second token was not tilted toward pair i, or when both tokens were committed in the same step. Three cases cover these outcomes.

(i) *The second token was drawn from its context p.* It lands in c with probability $m_c(p) = 1/q$ by (B).

(ii) *The second token was tilted toward a class s that completes a different pair.* By (U′), s is uniform given the history, the coloring, and c. By (B), the tilt toward s puts mass on c equal to

$$m_c(\mathsf{T}_s p) \;=\; \begin{cases} \dfrac{1+\gamma}{q+\gamma} & \text{if } s = c, \\[2ex] \dfrac{1}{q+\gamma} & \text{if } s \neq c, \end{cases} \qquad \text{so} \qquad \frac{1}{q}\cdot\frac{1+\gamma}{q+\gamma} + \frac{q-1}{q}\cdot\frac{1}{q+\gamma} \;=\; \frac{1}{q}.$$

(iii) *Both tokens of the pair were committed in the same step.* Then neither was tilted toward pair i, because a tilt toward a pair requires its other token to be committed already. Each token was drawn from its context or from a tilt toward a different pair, whose favored class is uniform by (U′). In both cases its color is uniform, by (B) and, for a tilt, by Lemma 2. By (S), the two candidates are independent given the state and the favored classes, and by (U′) the two favored classes are independent. The two colors are therefore independent and uniform, and

$$\Pr(R_i = b \mid \text{state}) \;=\; \sum_{u\in\mathbb{Z}_q} \Pr\big(\chi(t_{i-\delta}) = u\big)\,\Pr\big(\chi(t_i) = b - a_\delta u\big) \;=\; \sum_{u\in\mathbb{Z}_q} \frac{1}{q}\cdot\frac{1}{q} \;=\; \frac{1}{q}.$$

Step 3: averaging over enforcement. By Steps 1 and 2,

$$\mu_i \;=\; \frac{1}{q} + \Pr(\text{pair } i \text{ is enforced})\,\bar{\varepsilon}_{\text{enf}}.$$

Averaging over $i \in \mathcal{T}$ and using the definition of κ gives $\varepsilon = \kappa\,\bar{\varepsilon}_{\text{enf}}$. □

The factorization explains why enforcing from either side helps. At $\beta = 6$, either-side enforcement raises κ from 0.62 to 0.75 without changing the tilt at any position (Table 19).

A.4 Finite-sample error bounds

Proposition 2 gives the expected score but bounds no error probability. Under (I) the match indicators are independent variables in $\{0, 1\}$, and Hoeffding's inequality turns the expected separation into bounds on both error rates.

Corollary 2 (Finite-sample error bounds). *Assume (I). For $T \geq T_\star$ the false-negative rate satisfies*

$$\Pr(z \leq z_\star) \;\leq\; \exp\!\big(-2\big(\sqrt{T}\,\varepsilon - z_\star\sigma_0\big)^2\big).$$

Under the null, where every match probability is $1/q$, the false-positive rate satisfies, for every T,

$$\Pr(z > z_\star) \;\leq\; \exp\!\big(-2z_\star^2\sigma_0^2\big).$$

Proof. Under (I), Hoeffding's inequality (Hoeffding, 1963) gives, for every $\zeta \geq 0$,

$$\Pr\big(C - \mathbb{E}[C] \leq -\zeta\big) \;\leq\; e^{-2\zeta^2/T}, \qquad \Pr\big(C - \mathbb{E}[C] \geq \zeta\big) \;\leq\; e^{-2\zeta^2/T}.$$

False negatives. Since $\mathbb{E}[C] = T/q + T\varepsilon$,

$$\{z \leq z_\star\} \;=\; \big\{C \leq T/q + z_\star\sigma_0\sqrt{T}\big\} \;=\; \big\{C - \mathbb{E}[C] \leq -\zeta\big\}, \qquad \zeta \;=\; T\varepsilon - z_\star\sigma_0\sqrt{T}.$$

For $T \geq T_\star$ we have $\zeta \geq 0$, and $\zeta^2/T = (\sqrt{T}\varepsilon - z_\star\sigma_0)^2$. The first inequality gives the bound.

False positives. Under the null $\mathbb{E}[C] = T/q$, so

$$\{z > z_\star\} \;=\; \big\{C - \mathbb{E}[C] > z_\star\sigma_0\sqrt{T}\big\}.$$

The second inequality with $\zeta = z_\star\sigma_0\sqrt{T}$ gives $\exp(-2z_\star^2\sigma_0^2)$. □

For $T \geq T_\star$, the false-negative bound decays exponentially in T. The false-positive bound depends only on $z_\star$ and q. At $q = 3$ and $z_\star = 4$ it is $\exp(-64/9) = 8.2 \times 10^{-4}$. Hoeffding's inequality ignores the variance of the matches, so when the binomial null of Proposition 1 holds, the exact binomial tail is smaller.

Rounding in the class sizes moves each null match probability by at most $1/V$ (Section A.2), so it moves the null mean of C by at most T/V. Applying Hoeffding's inequality around the shifted mean changes the false-positive bound by less than 10^{-6} at $V \approx 1.26 \times 10^5$ and $T \leq 128$.

A.5 The count test is the likelihood-ratio test

The main text motivates the count C as the natural statistic for a checksum that holds at rate $1/q$ by chance. This subsection shows that, under three approximations and with one-sided enforcement, thresholding C is the likelihood-ratio test. By the Neyman–Pearson lemma, that test detects the largest fraction of watermarked texts among all detectors with the same false-positive rate.

Given a text $t_1, \ldots, t_L$, the optimal detector computes the likelihood ratio

$$\Lambda \;=\; \frac{\Pr(\text{text} \mid \text{watermarked model})}{\Pr(\text{text} \mid \text{no watermark})} \tag{6}$$

and flags the text when Λ exceeds a threshold. Computing Λ is infeasible. The numerator sums over every prompt and every unmasking order, both terms require running the model, and for human authors the denominator is unknown. Three approximations reduce Equation (6) to TANGO's detector. We index each scored pair by its scored position i, so that M_i is its match indicator. For an enforced pair i (Section A.1) we write $m_i = m_{s_i}(p)$ for the probability that the unwatermarked context p of the pair's tilted token assigns to the favored class.

Approximation 1: fix the prompt and the unmasking order. Assume that the watermarked text and its unwatermarked counterpart come from the same model, prompt, and unmasking order. The two hypotheses then differ only at the tilted token of each enforced pair. Consider one enforced pair i and its tilted token. Token probabilities are a softmax of logits ℓ_v, $p(v) = e^{\ell_v}/\Omega$ with $\Omega = \sum_w e^{\ell_w}$. The watermark adds β to the logits of the favored class. We first compute how this changes the normalizer, because the likelihood ratio of the token is its boost divided by that change. The new normalizer Ω' satisfies

$$\frac{\Omega'}{\Omega} \;=\; e^{\beta} \sum_{w:\, \chi(w)=s_i} p(w) \;+\; \sum_{w:\, \chi(w)\neq s_i} p(w) \;=\; 1 + \gamma\, m_i.$$

Given the contexts that this approximation fixes, both hypotheses assign the text a product of per-position token probabilities, and Approximation 3 reuses this factorization. The log-ratio $\log \Lambda$ is therefore a sum of per-token log-ratios. A token that was not tilted contributes 0, because both hypotheses use the same distribution there. The tilted token of an enforced pair lands in the favored class exactly when the pair's checksum holds. Its probability is multiplied by e^{β} in that case and in both cases divided by $1 + \gamma m_i$, so its log-ratio is

$$\log \frac{\Pr_{\text{wm}}(t)}{\Pr_0(t)} \;=\; \begin{cases} \beta - \log(1+\gamma m_i) & \text{if } M_i = 1, \\ -\log(1+\gamma m_i) & \text{if } M_i = 0. \end{cases}$$

Summing over enforced pairs,

$$\log \Lambda \;=\; \sum_{i \text{ enforced}} \Big[\beta M_i \;-\; \log\big(1 + \gamma\, m_i\big)\Big]. \tag{7}$$

Approximation 2: balance. The indicators M_i in Equation (7) depend only on the text and the key, but m_i requires running the model. Under (B), $m_i = 1/q$ at every position, and every penalty becomes the constant $\log(1 + \gamma/q)$.

Approximation 3: enforcement at random. The final text does not record the order in which positions were unmasked, so it does not reveal which pairs were enforced. Assume one-sided enforcement, so that the tilted token of pair i is t_i. Assume further that each scored pair was enforced with probability κ, independently across pairs. This independence is itself an approximation, since pairs i and $i + \delta$ share the token t_i. Under the null the factorization of Approximation 1 makes the colors of different positions independent and, by (B), uniform, which implies (I). Under the watermark, the probability of t_i is a mixture of its tilted and its unwatermarked probability,

$$\Pr_{\text{wm}}(t_i = v) \;=\; \kappa\, (\mathsf{T}_{s_i} p)(v) + (1-\kappa)\, p(v).$$

Under (B) the normalizer of every tilt is $1 + \gamma/q$. Dividing by $p(v)$ therefore cancels the token's identity and leaves only its match indicator, which gives the per-pair likelihood ratio

$$\lambda(M_i) \;=\; \frac{\Pr_{\text{wm}}(t_i = v)}{p(v)} \;=\; \kappa\, \frac{e^{\beta M_i}}{1 + \gamma/q} \;+\; (1 - \kappa). \tag{8}$$

Since $e^\beta = 1 + \gamma > 1 + \gamma/q$, Equation (8) gives $\lambda(1) > 1$ for a match and $\lambda(0) < 1$ for a miss. Both likelihoods factor over positions, so Λ is the product of the per-pair ratios, and over the T scored pairs

$$\log \Lambda \;=\; \underbrace{\log \frac{\lambda(1)}{\lambda(0)}}_{>0} \cdot C \;+\; T \log \lambda(0). \tag{9}$$

Proposition 4 (The count test is the likelihood-ratio test). *Under Approximations 1–3, which include one-sided enforcement, and among texts with the same number T of scored positions, $\log \Lambda$ is an increasing affine function of the match count C. Thresholding Λ is therefore the same test as thresholding C, or equivalently z. The threshold on C that gives a chosen false-positive rate depends only on the null, so the same count test is the likelihood-ratio test for every $\beta > 0$ and every $\kappa \in (0, 1]$.*

Proof. For fixed T, Equation (9) is affine in C with slope $\log(\lambda(1)/\lambda(0)) > 0$. Hence $\Lambda > \lambda_\star$ holds if and only if $C > C_\star$, for a threshold $C_\star$ determined by $\lambda_\star$, and the two tests flag the same texts. Under the null of Approximations 2 and 3, $C \sim \mathrm{Binom}(T, 1/q)$, which does not involve β or κ. The threshold $C_\star$ that gives a chosen false-positive rate is therefore the same for every β and κ. □

Because C is discrete, the Neyman–Pearson optimality holds exactly at the false-positive rates that a threshold on C attains. Other rates need a randomized threshold.

The proposition does not extend to either-side enforcement. There a token can be tilted toward either of its two pairs, so its likelihood ratio depends on both match indicators M_i and $M_{i+\delta}$, and $\log \Lambda$ is no longer a function of C alone. We use the count test for both modes.

Approximation 1 fits the question a provider asks about its own output, namely whether its model applied the bias while generating this text. Approximations 2 and 3 are (B) and (I), plus one-sided enforcement and the assumption that every pair is enforced with the same probability κ. Under one-sided enforcement the measured κ is 0.61 at $\beta = 8$ and 0.62 at $\beta = 6$ (Table 19). None of the three approximations enters the calibration of Proposition 1, so the false-positive rate can be set without the model.

A.6 Token frequencies and the key

An attacker who compares token frequencies in watermarked and unwatermarked text learns the key if the watermark shifts frequencies differently for different classes. This subsection proves that balanced tilts cancel in expectation (Lemma 2 and Theorem 1), bounds the shift when balance fails (Proposition 5), and shows that the red–green list has no such cancellation (Proposition 6).

The first two results rest on one identity. If a context spreads its mass evenly over the q classes, then tilting toward each class in turn and averaging returns the context.

Lemma 2 (Mixture identity). *If $m_s(p) = 1/q$ for every s, then $\frac{1}{q}\sum_{s=0}^{q-1} \mathsf{T}_s p \;=\; p$ for every β.*

Proof. Under the hypothesis every normalizer equals $1 + \gamma/q = (q + \gamma)/q$. Fix a token v. Exactly one of the q tilts multiplies $p(v)$ by e^β, and the other $q - 1$ leave it unchanged, so

$$\frac{1}{q}\sum_s (\mathsf{T}_s p)(v) \;=\; \frac{p(v)}{q} \cdot \frac{q}{q+\gamma}\big(e^\beta + q - 1\big) \;=\; p(v),$$

since $e^\beta + q - 1 = q + \gamma$. □

Theorem 1 (restated). *Fix a coloring χ. Under (S), (U), (B), the generated text has the same distribution with and without the watermark. Hence $\mathbb{E}[\Delta f(v)] = 0$ for every $v \in \mathcal{V}$, and any attacker who ranks each token v by $g(\mathbb{E}[\Delta f(v)])$, for a fixed function g, separates the color classes no better than chance (AUC $\frac{1}{2}$).*

Here the expectation is over the sampler and the favored classes of (U), with the coloring fixed. The proof must compare two whole generation processes, not one position. The model conditions on every committed token, so a watermark that changed which tokens appear early would also change the contexts of later positions. We therefore show that one denoising step has the same transition law with and without the watermark, given the whole history, and conclude by induction.

Proof of Theorem 1. Fix the coloring χ and the prompt. Write $h_j = (x^{(0)}, \dots, x^{(j)})$ for the history of states after j steps, and P_j and P_j^0 for the laws of h_j with and without the watermark.

Step 1: one step has the same transition law. Fix a history h_{j-1} with current state $x = x^{(j-1)}$. By (S), the candidates at the masked positions are drawn independently given x and the favored classes of the step, from $\mathsf{T}_{s_i} p_i$ at enforced positions and from p_i elsewhere, where the contexts p_i depend only on x. By (U), the favored classes of the enforced positions are independent and uniform given h_{j-1} and χ. Averaging over them, each enforced candidate has distribution

$$\frac{1}{q} \sum_{s \in \mathbb{Z}_q} \mathsf{T}_s p_i \;=\; p_i,$$

where the equality is Lemma 2 and is the only place (B) enters. Because the favored classes are independent across positions, averaging over them keeps the candidates independent, so given h_{j-1} they have the joint distribution $\prod_i p_i$, which is the unwatermarked one. By (S) the commit rule is also the same, so $x^{(j)}$ has the same law given h_{j-1} in both processes.

Step 2: induction over steps. Both processes start from the fully masked state, so $P_0 = P_0^0$. If $P_{j-1} = P_{j-1}^0$, then Step 1 shows that the two processes extend each history h_{j-1} by the same law, so $P_j = P_j^0$. By induction the laws of the full history agree, and in particular the final state, which is the generated text, has the same distribution in both processes. This holds for every prompt.

Step 3: frequencies and ranking. Every token therefore has the same expected frequency in both processes, and $\mathbb{E}[\Delta f(v)] = 0$ for every v. The score $g(\mathbb{E}[\Delta f(v)]) = g(0)$ therefore takes the same value on every token. A constant score ties every pair of tokens from different classes, and with ties counted as one half, the AUC is $\frac{1}{2}$. □

The theorem is a statement about the idealized sampler of (U), in which the favored class is a fresh draw at every position (Section A.1). It does not say that one key produces unwatermarked text, which would contradict detection. What it isolates is that each tilt, averaged over a favored class that is uniform and unrelated to the token's color, leaves the distribution unchanged. For a real key, frequency shifts can therefore arise only where the favored class is related to the color of the token drawn, which happens when the taps' colors are unevenly spread over the classes or correlated with the color at the tilted position.

The contrast with the red–green list does not come from this theorem. That list favors the same class at every position, so the favored class never varies and no sampler of the kind (U) describes approximates it. Proposition 6 shows that its frequency shifts then separate the green list from the rest.

The theorem also concerns expected frequencies only. An attacker sees a finite number of texts, whose empirical frequency shifts are noisy and need not vanish exactly. The theorem does not cover this attacker, and Table 8 measures what it recovers.

Frequency shifts when balance fails. Theorem 1 rests on (B). When a context puts more than $1/q$ of its mass on some class, the tilts no longer cancel exactly and token frequencies shift. The next proposition gives the shift in closed form and bounds it by the imbalance.

Proposition 5 (Imbalance bounds the frequency shift). *Fix a context p with class masses $m_s = m_s(p)$ and imbalances $\eta_s = m_s - 1/q$, and let $\phi(x) = 1/(1+\gamma x)$. For every token v with $\chi(v) = c$,*

$$\frac{1}{q} \sum_s (\mathsf{T}_s p)(v) - p(v) \;=\; p(v)\, \frac{\gamma}{q} \Big[\phi(m_c) - \sum_s m_s \phi(m_s) \Big],$$

and its magnitude is at most $p(v)\, \frac{2\gamma^2}{q} \max_s |\eta_s|$. To first order in the imbalance, the shift is $-p(v)\, \gamma^2 \eta_c / (q \bar{Z}^2)$ with $\bar{Z} = 1 + \gamma/q$, the normalizer under balance.

Proof. Step 1: the identity. The tilt toward v's own class multiplies $p(v)$ by $(1+\gamma)\phi(m_c)$, and the tilt toward any other class s multiplies it by $\phi(m_s)$, so

$$\frac{1}{q} \sum_s (\mathsf{T}_s p)(v) \;=\; \frac{p(v)}{q} \Big[(1+\gamma)\phi(m_c) + \sum_{s \neq c} \phi(m_s) \Big] \;=\; \frac{p(v)}{q} \Big[\gamma \phi(m_c) + \sum_s \phi(m_s) \Big].$$

From $\phi(x)(1+\gamma x)=1$ and $\sum_s m_s = 1$,

$$\phi(m_s) \;=\; 1-\gamma\, m_s\phi(m_s) \quad\Longrightarrow\quad \sum_s \phi(m_s) \;=\; q-\gamma\sum_s m_s\phi(m_s).$$

Substituting into the bracket and subtracting $p(v)$ proves the identity.

Step 2: the bound. We compare ϕ at m_c with ϕ at each m_s through the mean value theorem. Since $\sum_s m_s = 1$,

$$\phi(m_c) - \sum_s m_s\phi(m_s) \;=\; \sum_s m_s\big(\phi(m_c)-\phi(m_s)\big).$$

On $[0,1]$ we have $|\phi'(x)| = \gamma/(1+\gamma x)^2 \le \gamma$, so

$$|\phi(m_c)-\phi(m_s)| \;\le\; \gamma\,|\eta_c-\eta_s| \;\le\; 2\gamma\max_s|\eta_s|.$$

Averaging over s with weights m_s and multiplying by $p(v)\gamma/q$ gives the bound.

Step 3: the first-order term. Expanding around balance,

$$\phi(m_s) \;=\; \phi(1/q)+\phi'(1/q)\,\eta_s+O(\eta^2), \qquad \phi'(1/q) \;=\; -\gamma/\bar{Z}^2.$$

Since $\sum_s \eta_s = 0$, we get $\sum_s m_s\phi(m_s) = \phi(1/q)+O(\eta^2)$, so the bracket equals

$$\phi'(1/q)\,\eta_c + O(\eta^2) \;=\; -\gamma\,\eta_c/\bar{Z}^2 + O(\eta^2).$$

Multiplying by $p(v)\gamma/q$ gives the stated term. □

The shift is first order in the imbalance, and for small γ it is second order in γ. To first order, it depends only on the imbalance of v's own class. A token in an over-represented class ($\eta_c > 0$) becomes slightly less frequent under the watermark, and a token in an under-represented class slightly more frequent. At the default $\beta = 5$, where $\gamma \approx 147$, the two coefficients are

$$\frac{2\gamma^2}{q} \;\approx\; 1.4\times10^4 \quad\text{(uniform bound)}, \qquad \frac{\gamma^2}{q\bar{Z}^2} \;\approx\; 2.9 \quad\text{(first order)}.$$

The uniform bound exceeds $p(v)$ at this bias, so the first-order coefficient is the informative one. It says that a class imbalance of 0.06 moves a token's expected frequency in one context by about 17% of $p(v)$, in a direction set by the key. The evidence that this shift does not reveal the key in practice comes from the key-recovery experiment of Table 8.

The red–green list. The red–green list tilts every position toward the same token set $G \subset \mathcal{V}$, so there is no average over favored classes and nothing cancels. To isolate this effect, we hold the contexts fixed and compare the tilted and untilted distributions in each one. Like the tilt toward a class in Equation (4), the tilt toward G multiplies the probability of each token in G by e^β and renormalizes. Let $m_G(p) = \sum_{v\in G} p(v)$ be the total probability that p assigns to tokens in G. The tilt is

$$(\mathsf{T}_G p)(v) \;=\; \frac{p(v)\,e^{\beta\mathbf{1}[v\in G]}}{1+\gamma\,m_G(p)}.$$

For contexts $p_1,\dots,p_H$, the average shift is

$$\Delta_G f(v) \;=\; \frac{1}{H}\sum_{h=1}^{H}\Big[(\mathsf{T}_G p_h)(v)-p_h(v)\Big].$$

Proposition 6 (The red–green list shifts frequencies toward its key). *If $0 < m_G(p_h) < 1$ for every context, and every token has $p_h(v) > 0$ in at least one context, then $\Delta_G f(v) > 0$ for every $v \in G$ and $\Delta_G f(v) < 0$ for every $v \notin G$. Ranking tokens by $\Delta_G f$ therefore separates G from its complement with AUC 1.*

Proof. In context p_h the normalizer is $Z_h = 1+\gamma\,m_G(p_h)$, which lies strictly between 1 and $1+\gamma = e^\beta$ because $0 < m_G(p_h) < 1$. The shift in context h is

$$(\mathsf{T}_G p_h)(v)-p_h(v) \;=\; \begin{cases} p_h(v)\,\big(e^\beta/Z_h-1\big) & \text{if } v\in G, \text{ with } e^\beta/Z_h > 1,\\ p_h(v)\,\big(1/Z_h-1\big) & \text{if } v\notin G, \text{ with } 1/Z_h < 1.\end{cases}$$

Each shift is therefore nonnegative for $v \in G$ and nonpositive for $v \notin G$, and it is nonzero in every context with $p_h(v) > 0$. Averaging over contexts, $\Delta_G f(v) > 0$ on G and $\Delta_G f(v) < 0$ off G. Every token of G then ranks above every token outside G, so every pair from different sets is ordered correctly and the AUC is 1. □

Holding the contexts fixed isolates the direct effect of the tilt. In real generation the tilt also changes later contexts, which this statement does not model. Figure 2 shows that the effect is visible in practice.

A.7 Edit robustness

Theorem 2 models an attacker who substitutes tokens at random. It uses only the following attack model and none of (S), (U), (U′), (B), or (I).

- The attacker draws corruption indicators $X_j \sim \mathrm{Bern}(\rho)$ independently across positions and independently of the text.
- A corrupted token's color is replaced by a variable ξ_j that is uniform on $\mathbb{Z}_q$ and independent of everything else.
- The attacker neither inserts nor deletes tokens, so every position keeps its index.
- The residue b is fixed, and $\gcd(a_\delta, q) = 1$ for every $\delta \in \mathcal{D}$.

The proof counts which checksums an edit touches. A checksum reads $|\mathcal{D}| + 1$ tokens. If none of them is corrupted, the checksum behaves as on clean text. If any is corrupted, it matches at exactly the chance rate. The excess over chance therefore survives with the probability that all $|\mathcal{D}| + 1$ tokens are untouched.

Theorem 2 (restated). *Under the attack model above,* $\mathbb{E}[z_{\mathrm{att}}] = (1-\rho)^{|\mathcal{D}|+1}\,\mathbb{E}[z_{\mathrm{clean}}]$*, where* z_{att} *and* z_{clean} *are the scores of the attacked and the clean text.*

Proof of Theorem 2. Fix a scored position i and the set of positions its checksum reads,

$$A_i \;=\; \{i\} \cup \{i - \delta : \delta \in \mathcal{D}\},$$

which has $|\mathcal{D}| + 1$ elements. Condition on the corruption pattern $(X_j)_{j \in A_i}$, and write χ' for the colors after the attack. We show that

$$\Pr\left(M_i = 1 \mid (X_j)_{j \in A_i}\right) \;=\; \begin{cases} \mu_i & \text{if no position in } A_i \text{ is corrupted,} \\ 1/q & \text{otherwise.} \end{cases}$$

Case 1: no position in A_i is corrupted. This has probability $(1-\rho)^{|\mathcal{D}|+1}$, and the residue is computed on clean tokens. The corruption pattern is independent of the text, so the conditional match probability is the clean μ_i.

Case 2: at least one position in A_i is corrupted. The attacked residue is

$$R_i \;=\; \Big(\chi'(t_i) + \sum_{\delta \in \mathcal{D}} a_\delta\, \chi'(t_{i-\delta})\Big) \bmod q.$$

Pick one corrupted position $j \in A_i$. Its term in R_i is ξ_j if $j = i$, and $a_\delta \xi_j$ if $j = i - \delta$. In the second case $a_\delta \xi_j$ is uniform, because multiplication by a_δ permutes $\mathbb{Z}_q$ when $\gcd(a_\delta, q) = 1$. Write $R_i = (w_j + y_j) \bmod q$, where w_j is this term and y_j is the sum of the remaining terms, which involves only the text and the replacement colors of the other positions. The term w_j is uniform and independent of y_j. The sum of a uniform variable and an independent variable on $\mathbb{Z}_q$ is uniform, so R_i is uniform and $\Pr(R_i = b) = 1/q$.

Averaging. Averaging over the corruption pattern and summing over positions,

$$\mathbb{E}[C_{\mathrm{att}}] \;=\; \sum_{i \in \mathcal{T}} \Big[(1-\rho)^{|\mathcal{D}|+1}\mu_i + \big(1 - (1-\rho)^{|\mathcal{D}|+1}\big)\tfrac{1}{q}\Big] \;=\; \tfrac{T}{q} + (1-\rho)^{|\mathcal{D}|+1}\, T\varepsilon.$$

Substitutions do not change T, so the same $\sqrt{T}\,\sigma_0$ normalizes both scores. Applying Proposition 2 to the clean and attacked texts gives $\mathbb{E}[z_{\mathrm{att}}] = (1-\rho)^{|\mathcal{D}|+1}\,\mathbb{E}[z_{\mathrm{clean}}]$. □

The condition $\gcd(a_\delta, q) = 1$ is used only for a corrupted tap. If $g = \gcd(a_\delta, q) > 1$, then $a_\delta \xi_j$ takes only the q/g multiples of g in $\mathbb{Z}_q$, and a checksum whose only corrupted position is that tap need not match at rate $1/q$. When q is prime, every nonzero coefficient satisfies the condition.

Insertions and deletions. These edits shift positions, so an edit at position j changes which tokens form the pairs that span j. The theorem does not model this. For these edits it predicts the direction of the effect (Figure 3) but not its size.

Synonyms. The theorem gives a corrupted token a uniformly random color. The semantic-quantile coloring is designed so that a synonym keeps its color more often than chance. To the extent that it does, the theorem underestimates how much of the score survives synonym substitution.

B Experimental details

This appendix gives the settings shared by all experiments, the two perplexity protocols, the attacks, and the rule that selected TANGO's defaults. Settings that differ for a single experiment are stated in the caption of its table.

Decoding. All methods generate 128 tokens in 128 denoising steps at temperature 1.0 with no top-p truncation. On LLaDA every method uses classifier-free guidance (CFG) at scale 2.0. Dream's sampler has no CFG, so Dream runs use a plain forward pass. These settings are the same for every method on a model, so perplexities are comparable within a model. Coloring and scoring use each model's full tokenizer vocabulary ($V = 126{,}464$ for LLaDA and $152{,}064$ for Dream).

Perplexity. The main text reports perplexity under Qwen2.5-7B-Instruct in bfloat16, conditioned on the prompt with a BOS token prepended. We pool the continuations into one perplexity, the exponential of their total negative log-likelihood divided by their total number of tokens. Appendix columns marked PPL GPT-2 report the GPT-2-large (Radford et al., 2019) perplexity of the continuation alone, averaged over texts, and columns marked PPL Qwen report the pooled Qwen perplexity.

Degenerate continuations. Perplexity rewards repetition, because each repeat of a phrase is near-certain given the earlier copies. We call a continuation *degenerate* when more than 20% of its whitespace 4-grams repeat an earlier 4-gram of the same text. In practice this means a phrase or a number repeated up to the length limit. Tables 1, 12, and 13 score perplexity over the non-degenerate continuations only, and every other appendix table scores all continuations. On LLaDA the choice matters little. At the settings of Tables 1 and 12, at most 6% of texts are degenerate. The two perplexities differ by at most 0.5 at the settings of Table 1 and by at most 1.3 across Table 12. The ablation settings of Section E reach 12.5%. On Dream the choice matters more. There, 20% of unwatermarked texts, 31% of TANGO texts, and 62% of red–green texts are degenerate. Over all continuations, the perplexities behind Table 1 are 4.1 (unwatermarked), 7.0 (TANGO), and 7.4 (red–green list) on LLaDA, and 3.5, 6.2, and 4.9 on Dream. Repeated phrases have low perplexity, so scoring all continuations favors the method that degenerates most often, which on Dream is the red–green list.

Attacks and metrics. The attacks are word-level edits from the MarkLLM toolkit (Pan et al., 2024). They delete 10% or 30% of the words (del10, del30), replace 30% or 50% of the words with WordNet synonyms (syn30, syn50), or insert random words amounting to 20% of the text (ins20). The main text reports del30, syn30, and ins20. TPR@1%FPR sets the threshold at the 99th percentile of the detector scores of the unwatermarked texts listed in each caption. With fewer than 100 such texts, this threshold is close to their maximum score, and these texts cannot certify a false-positive rate as low as 1%. Bootstrap intervals use 1000 resamples of the generations.

How the defaults were chosen. The generation settings were chosen in two stages. First, the ablations of Section E, run with one-sided enforcement at $\beta = 8$, fixed the coloring. We use $q = 3$ because $q = 2$ detects less under edits and larger q costs perplexity (Table 17). We use a single tap

Algorithm 1 One denoising step of TANGO generation. On Dream, which has no CFG, the logits come from a plain forward pass. a_δ^{-1} is the inverse of a_δ modulo q, which exists because $\gcd(a_\delta, q) = 1$.

Require: key-derived coloring χ, tap lag δ, coefficient a_δ with $\gcd(a_\delta, q) = 1$, residue b; logit bias β; temperature τ; unmasking schedule $K(\cdot)$

$\boldsymbol{\ell} \leftarrow$ CFG logits for all masked positions ▷ one bidirectional forward pass
for each masked position i in the step's candidate set **do**
  if $t_{i-\delta}$ is unmasked **then**
    $s_i \leftarrow \big(b - a_\delta\, \chi(t_{i-\delta})\big) \bmod q$ ▷ left tap at lag δ unmasked
  else if $t_{i+\delta}$ is unmasked **then**
    $s_i \leftarrow a_\delta^{-1}\big(b - \chi(t_{i+\delta})\big) \bmod q$ ▷ right neighbor at lag δ unmasked
  else
    continue ▷ neither unmasked, so no bias
  end if
  $\ell_{i,v} \leftarrow \ell_{i,v} + \beta$ **for all** v **with** $\chi(v) = s_i$
end for
sample one candidate per masked position at temperature τ; unmask the K most confident (schedule on remaining steps)

because every extra tap exposes the checksum to more edits (Theorem 2, Table 18). We use $b = 0$ because it has the lowest null rate on natural text under the default key (Section C.3).

Second, a selection experiment chose the enforcement side and the bias. It used 48 held-out C4 prompts (prompts 400 to 447, disjoint from the 200 evaluation prompts), and its selection rule was fixed before it ran. The rule compared each setting with a reference, one-sided enforcement at $\beta = 8$, which biases a position only when its left tap is unmasked. A setting was eligible if its clean TPR was at least 0.98 and its del30 TPR was at least the reference's. On 48 prompts the first condition requires detecting every text, so 47 of 48, printed as 0.98, does not qualify. An eligible setting was then excluded if its success at forgery without a language model (Table 9) exceeded the reference's by more than 0.10. Among the remaining settings, the rule took the one with the lowest Qwen perplexity. It selected either-side enforcement at $\beta = 5$ (Qwen perplexity 7.0). Table 19 lists all 14 settings, and Section E.3 discusses them. On the 200 evaluation prompts the selected setting detects 68% of deleted texts, against 79% on the 48 selection prompts, a gap within the noise of 48 prompts. For this reason Table 12 compares methods over the full range of β and does not rely on one operating point.

Computational cost. Generation adds one logit update over the vocabulary per eligible position per denoising step, an $O(V)$ cost next to the $O(Vd)$ output layer of the forward pass, where d is the embedding dimension. The coloring is built once per key. Building it projects the V token embeddings onto the key direction at cost $O(Vd)$ and sorts them at cost $O(V \log V)$. Detection then scores a text in one $O(L)$ pass over its L tokens.

Software, models, and data. The code runs on Python 3.12 with a CUDA 12 build of PyTorch and Transformers at least 4.46 and below 4.57, and `requirements.txt` in the repository lists the other packages. Perplexity is scored with Qwen/Qwen2.5-7B-Instruct and gpt2-large. The removal attack of Section D rewrites words with distilroberta-base, and the paraphrase translates with Helsinki-NLP/opus-mt-en-zh and opus-mt-zh-en. The edit attacks are adapted from MarkLLM and draw synonyms from NLTK's WordNet. Prompts and natural texts come from the processed C4 file in the repository. The evaluation prompts are its first 200 records, and the Dream runs use the first 100. The held-out selection prompts are records 400 to 447, and the forging attacker of Table 10 generates on records 200 to 299. Bootstrap intervals use seed 0. The repository caches the colorings used in the paper, so every detection score can be recomputed from the released texts without model weights. The published diffusion watermarks run from their public code at pinned commits, through drivers in the repository.

C Calibration of the detector on unwatermarked text

The threshold $z_\star$ controls false positives only if the null of Proposition 1 describes unwatermarked text. That proposition assumes that the scored token's color is uniform and independent of its tap's color, and that matches at different positions are independent, which (B) and (I) idealize. This appendix checks how far unwatermarked text from both models departs from these assumptions, and where the departure comes from. It supports the per-model calibration of the threshold in Sections 3.1 and 3.3.

C.1 LLaDA

On LLaDA the fixed threshold $z > 4$ flags no unwatermarked text. We score two kinds of unwatermarked text, 500 texts each. Text of uniformly random tokens satisfies the hypotheses of Proposition 1 by construction, and its score is close to standard normal (mean -0.07, standard deviation 0.99). In natural C4 text the colors of nearby tokens are correlated (Section C.3). Its score is shifted left (mean -0.46, standard deviation 1.08), so the threshold flags natural text less often than the normal tail predicts. With 0 of 500 texts above the threshold, the exact two-sided 95% Clopper–Pearson upper bound on the false-positive rate is 0.74% (Clopper & Pearson, 1934).

C.2 Dream

On Dream the random-token null is again close to standard normal, but natural text satisfies the constraint more often than chance. Its mean score is $+1.07$, and 1.00% of 300 natural texts score above 4 (Table 5). Because random tokens are calibrated, the excess comes from structure in natural text. The likely source is the correlation between the colors of neighboring tokens, which Section C.3 measures on LLaDA.

Building the coloring from whitened embeddings reduces the excess. ZCA whitening (Kessy et al., 2018) linearly transforms the embeddings so that their dimensions are uncorrelated with unit variance, and we call the coloring built from them the *whitened coloring*. On Dream it lowers the mean natural score from 1.07 to 0.67 and the rate above $z = 4$ from 1.00% to 0.33%. Whitening does not reduce detection. With one-sided enforcement at $\beta = 8$ ($n = 100$), del30 TPR is 0.82 with and without it, and syn30 and ins20 differ by 0.01. We therefore decide per model whether to whiten. The defaults on both models do not whiten, so the Dream rows of Tables 1 and 13 use the unwhitened coloring. Some LLaDA diagnostics in Sections D and E use the whitened coloring, as their captions state. A new model needs its threshold set on its own unwatermarked text.

Table 5: Whitening lowers the score of unwatermarked natural text on Dream, and random-token text stays calibrated with or without it. 300 texts per row. KS is the Kolmogorov–Smirnov distance to $\mathcal{N}(0, 1)$.

coloring	null source	mean z	std	KS	FPR $z{>}4$
unwhitened	natural	1.07	1.10	0.403	1.00%
unwhitened	random tokens	0.02	1.04	0.047	0.00%
ZCA	natural	0.67	1.17	0.274	0.33%
ZCA	random tokens	-0.05	1.00	0.057	0.00%

C.3 Correlated colors and the choice of residue

Natural text departs from the hypotheses of Proposition 1 in two ways, and both make the null depend on the residue b, a choice the theory says should not matter. First, the classes have equal numbers of tokens but not equal frequencies (the class masses in Table 6). Second, tokens that occur near each other have correlated embeddings and therefore correlated colors. Both effects enter through the $q \times q$ table $P(u, w)$ of how often a tap of color u precedes a token of color w. A position matches residue b when $w = (b - a_\delta u) \bmod q$, so each residue sums a different set of q cells,

$$\Pr(M_i = 1) \;=\; \sum_{u \in \mathbb{Z}_q} P\big(u, (b - a_\delta u) \bmod q\big), \qquad \text{which equals} \quad \sum_{u \in \mathbb{Z}_q} \pi_u \, \pi_{(b - a_\delta u) \bmod q}$$

when the two colors are independent with class masses $\pi_0, \ldots, \pi_{q-1}$. With unequal masses this rate already differs across residues. For the natural-text masses of Table 6 it is 0.329, 0.338, and 0.333 for $b = 0, 1, 2$. Correlated colors move the rates further.

Table 6 measures these rates for the default key on 2000 natural C4 texts and on the 200 unwatermarked LLaDA continuations of Table 1. On natural text at lag 1, each cell of the table differs from the product of its row and column frequencies by up to 0.013, and the null mean score ranges from -0.93 to $+0.98$ across residues. At the default lag 2 the difference is at most 0.005 per cell, and the range of null means is less than half as wide. This is why the default tap skips the adjacent token.

The residue changes the null more than the watermarked score (Table 7). The three residues give mean watermarked scores within 0.5 of each other but null means that differ by up to 0.7, and TPR at a fixed false-positive rate depends on where the null sits. The residue with the lowest null is a property of the key. Across eight fresh keys, it was $b = 2$ for six, $b = 0$ for one, and $b = 1$ for one. The residue should therefore be chosen, or the threshold calibrated, per key on unwatermarked text. A tap-keyed residue, set by a keyed hash of the tap color, avoids choosing b and lowers the null mean to -0.93. It detects slightly fewer unedited texts than $b = 0$ (0.97 against 1.00) and more deleted texts (0.91 against 0.79).

Table 6: Unwatermarked text satisfies each residue at a different rate under the default coloring ($q = 3$). *Class mass* is the fraction of natural-text tokens in each color class, whose token counts are equal. *Null rate* is the fraction of scored positions whose color pair satisfies the checksum for residue b (chance $1/3$). *Null mean* z is the mean detector score per text. *Natural* is 2000 C4 texts, and *model* is the 200 unwatermarked LLaDA continuations of Table 1.

text	lag	class mass ($0/1/2$)	null rate ($b = 0/1/2$)	null mean z ($b = 0/1/2$)
natural	1	$0.317/0.291/0.392$	$0.302/0.366/0.332$	$-0.93/+0.98/-0.05$
natural	2	$0.317/0.291/0.392$	$0.319/0.341/0.340$	$-0.42/+0.24/+0.19$
model	1	$0.326/0.263/0.411$	$0.298/0.362/0.340$	$-0.83/+0.68/+0.15$
model	2	$0.326/0.263/0.411$	$0.303/0.358/0.339$	$-0.71/+0.58/+0.13$

Table 7: The residues differ in the null score more than in the watermarked score. Lag 2, $a_2 = 1$, one-sided enforcement at $\beta = 8$, $n = 100$. *Mean* z is the mean score of unedited watermarked text, and *null mean* z the mean score of 500 natural texts (a different sample from the 2000 texts of Table 6). *Tap-keyed* sets the residue by a keyed hash of the tap color.

residue	AUROC	clean	mean z	null mean z	del30	syn30	ins20	PPL GPT-2
$b = 0$	1.000	1.00	6.33	-0.46	0.79	0.92	0.93	27.6
$b = 1$	0.999	0.97	6.59	$+0.24$	0.32	0.58	0.68	26.1
$b = 2$	1.000	1.00	6.80	$+0.22$	0.76	0.98	0.99	25.2
tap-keyed	0.991	0.97	6.02	-0.93	0.91	0.93	0.97	26.9

D Full results

This appendix gives the full tables behind Sections 4.2 and 4.3, the Dream results (Section D.5), and the attacks and robustness checks that the main text only mentions. Unless a caption says otherwise, results are on LLaDA with the 200 C4 evaluation prompts, the attack columns are TPR@1%FPR under del30, syn30, and ins20, and the negatives for TPR@1%FPR are the 200 unwatermarked LLaDA continuations of the same prompts (Section B). Several experiments use *one-sided* enforcement at $\beta = 8$, which biases a position only when its left tap is unmasked. This is the reference setting of the selection experiment (Section B). Some also use the whitened coloring (Section C). Each caption states its configuration.

D.1 Key recovery and forgery from token frequencies

The attacker of Theorem 1 ranks single tokens by how much more often they appear in watermarked than in unwatermarked text. Table 8 reports how much the token frequencies shift, how well the

shift predicts the key, and how often forgeries built from the predicted key pass the detector. Against TANGO the predicted key is at chance and no forgery passes. Against the red–green list the key is recovered with AUC 0.817 and every forgery passes. The table uses one-sided enforcement at $\beta = 8$. On the either-side texts of Table 1, key recovery against TANGO is also 0.504. Figure 2 shows that the red–green list's key becomes easier to recover as the attacker collects more texts, while TANGO's stays at chance, consistent with Theorem 1.

Table 8: Token frequencies reveal the red–green list's key but not TANGO's. 200 texts per method, TANGO with one-sided enforcement at $\beta = 8$, red–green list at bias 5. *Frequency shift* is the total-variation distance between watermarked and unwatermarked token frequencies (sampling noise alone gives 0.210). *Key recovery* is the AUC of predicting the coloring from per-token frequency shifts. *Forgery success* is the fraction of texts generated from the predicted key that the detector accepts at $z > 4$.

method	frequency shift (TV)	key recovery AUC	forgery success
TANGO	**0.217**	**0.504**	**0.00**
Red–green list	0.340	0.817	1.00

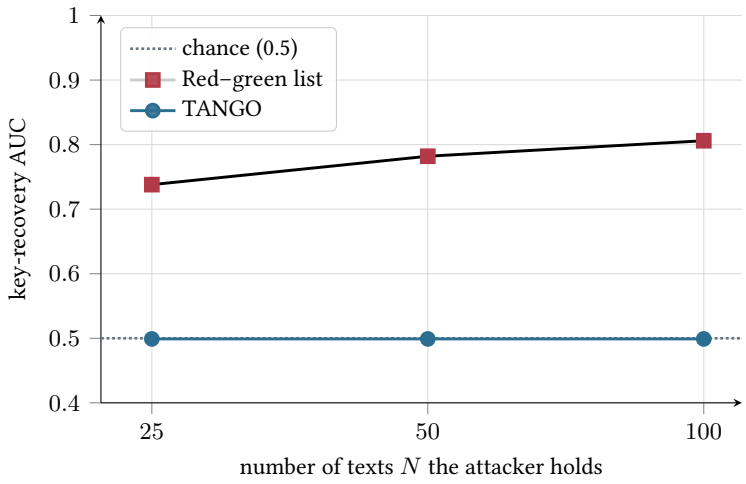

Figure 2: Key recovery from token frequencies improves with the number of texts against the red–green list (0.738 at 25 texts, 0.806 at 100) and stays at chance against TANGO. Subsets of a pool of 100 texts per method, TANGO with one-sided enforcement at $\beta = 6$ and a whitened coloring, red–green list at bias 5.

D.2 Key recovery and forgery from token pairs

Theorem 1 covers single-token frequencies only. This subsection measures an attacker it does not cover, one who counts token pairs at the tap lag. Table 9 reports two steps of this attack for every method. The first step ranks pairs by how much more often they appear in watermarked than in unwatermarked text. Its *pair AUC* measures how well this ranking predicts which pairs satisfy the checksum under the true key. The second step builds a text without a language model, by chaining pairs that are over-represented in watermarked text, and submits it to the detector. Against TANGO the pair AUC is 0.62, and 0.78 on pairs seen at least three times. At 200 texts these forgeries pass TANGO's detector 74% of the time, against 96% for the red–green list. Against dgMARK at its greedy defaults, no forgery passes at any N (mean score 1.4 at 200 texts), because those watermarked texts carry only a small excess of even token ids to learn from (Section D.3).

Text strung together without a language model is not fluent. Table 10 therefore reports a more realistic attacker, who has the model but not the key. While generating, it adds a logit bias β_{forge}, the forging bias. Against TANGO the bias favors the tokens that followed the tap token in the collected pairs, and against the red–green list it favors the recovered green list. At $\beta_{\text{forge}} = 4$,

its texts pass TANGO's detector 4% of the time and the red–green detector 51% of the time, at GPT-2-large perplexity 13.5 and 15.2 against 13.1 for unwatermarked text. At $\beta_{\text{forge}} = 8$ the rates are 55% and 98%, and the TANGO forgeries have perplexity 21.1, against 13.1 for unwatermarked text.

Table 9: Forgeries built from token-pair counts pass TANGO's detector less often than the red–green list's at every N. Each cell is *pair AUC / forgery success*, averaged over five random subsets of the N texts the attacker holds. Pair AUC uses each method's tap lag (2 for TANGO, 1 otherwise). Forgery success is the fraction of 50 forged 128-token texts accepted at $z > 4$. TANGO uses the either-side texts of Table 1. – marks detectors this attacker cannot query and dgMARK's pair AUC, which its keyless test leaves undefined (Section D.3). KGW is the context-hashed green list of Kirchenbauer et al. (2023).

model	method	$N = 25$	$N = 50$	$N = 100$	$N = 200$
LLaDA	TANGO	0.62 / 0.12	0.62 / 0.18	0.62 / 0.48	0.62 / 0.74
LLaDA	Red–green list	0.63 / 0.51	0.64 / 0.74	0.65 / 0.89	0.66 / 0.96
LLaDA	Gumbel	0.53 / 0.17	0.54 / 0.25	0.53 / 0.24	0.53 / 0.56
LLaDA	DLM watermark, defaults	0.54 / –	0.55 / –	0.54 / –	0.54 / –
LLaDA	DLM watermark, bias 2	0.53 / –	0.53 / –	0.55 / –	0.55 / –
LLaDA	DLM watermark, bias 3	0.55 / –	0.55 / –	0.56 / –	0.57 / –
LLaDA	DLM watermark, bias 4	0.57 / –	0.58 / –	0.59 / –	0.60 / –
LLaDA	KGW (context-hashed), bias 2	0.55 / –	0.55 / –	0.56 / –	0.56 / –
LLaDA	dgMARK, greedy defaults	–/ 0.00	–/ 0.00	–/ 0.00	–/ 0.00
LLaDA	dgMARK, top-k 3	–/ 0.09	–/ 0.21	–/ 0.26	–/ 0.44
Dream	TANGO	0.58 / 0.14	0.59 / 0.31	0.59 / 0.49	–
Dream	Red–green list	0.69 / 0.29	0.67 / 0.51	0.68 / 0.64	–

Table 10: At forging bias $\beta_{\text{forge}} = 4$, an attacker with the model but not the key passes TANGO's detector in 4% of cases and the red–green list's in 51%. The attacker mines the 200 watermarked texts of each method from Table 1 and generates on 100 unseen prompts. Against TANGO it biases each position toward the tokens that, in the collected lag-2 pairs, followed the token two positions to its left. Against the red–green list it biases every position toward the tokens most over-represented in watermarked text. Success is at each detector's empirical 1% FPR threshold on these prompts ($z = 1.96$ for TANGO, 4.15 for the red–green list) and at the fixed $z > 4$. PPL is GPT-2-large perplexity, and unwatermarked text on these prompts has PPL 13.1.

β_{forge}	detector	mean z	success (1% FPR)	success ($z > 4$)	PPL GPT-2
4	TANGO	0.08	0.04	0.00	13.5
4	Red–green list	4.15	0.51	0.56	15.2
8	TANGO	2.62	0.55	0.23	21.1
8	Red–green list	10.56	0.98	0.98	30.2

D.3 The published diffusion watermarks

Neither of the two published watermarks for masked-diffusion models that we ran detects more than TANGO or the red–green list at a similar perplexity cost (Table 11), which supports the comparison in Section 4.2. We ran the DLM watermark (Gloaguen et al., 2026) and dgMARK (Hong & No, 2026) from their public code on the same 200 prompts, at their own defaults and at our decoding settings. Their samplers differ from ours, so each perplexity is paired with an unwatermarked control from the same sampler. The best-detecting published settings are the DLM watermark at bias 4, which detects 64% of clean texts at about 1.14 times its control perplexity, and dgMARK with top-k sampling, which detects 84% at 2.2 times. The DLM watermark falls on the red–green list's curve in Table 12. That list detects 58% of clean texts at 1.1 times its control perplexity (bias 2) and 83% at 1.2 times (bias 3). dgMARK detects fewer clean texts than TANGO, which detects 97% at 1.7 times.

The released dgMARK detector needs no key. Its code derives the expected parity and the observed token parity from the same key bit, so the two cancel, and the test reduces to counting even token

ids. Any text rich in even token ids therefore passes it. The watermarked texts have 58% (greedy defaults) and 68% (top-k) even ids, against 48% for unwatermarked text, which is why its forgeries in Table 9 target even ids. We report dgMARK as released. Its authors suggest a keyed hash as future work.

Table 11: Neither of the two published diffusion watermarks we ran detects more than TANGO or the red–green list at a similar perplexity increase over its own control. 200 C4 prompts, 128 tokens, LLaDA-8B. TANGO, the red–green list, and the Gumbel rule are at the settings of Table 1. The published diffusion watermarks were run from their public code at their own defaults and at our decoding settings (*matched*). Each perplexity is followed, in parentheses, by that of unwatermarked text from the same sampler, and both are over all continuations. *Freq. AUC* is key recovery from token frequencies (Table 8). *Pair AUC* and *forge* are at $N = 200$ from Table 9, where KGW is also defined.

		TPR@1%FPR				PPL (control)				
method	AUROC	clean	del30	syn30	ins20	Qwen	GPT-2	freq. AUC	pair AUC	forge
TANGO ($\beta = 5$, either side)	0.998	0.97	0.68	0.92	0.91	7.0 (4.1)	18.9 (12.9)	0.50	0.62	0.74
Red–green list, bias 5	0.993	0.98	0.96	0.94	0.98	7.4 (4.1)	17.6 (12.9)	0.82	0.66	0.96
Gumbel (matched)	0.815	0.15	0.10	0.07	0.07	4.5 (4.1)	13.6 (12.9)	–	0.53	0.56
DLM watermark, defaults (bias 2)	0.843	0.37	0.30	0.22	0.24	6.5 (6.1)	39.9 (71.1)	–	0.54	–
DLM watermark, matched, bias 2	0.808	0.19	0.16	0.09	0.09	4.2 (4.0)	24.1 (21.5)	–	0.55	–
DLM watermark, matched, bias 3	0.873	0.34	0.20	0.16	0.14	4.2 (4.0)	24.3 (21.5)	–	0.57	–
DLM watermark, matched, bias 4	0.915	0.64	0.44	0.32	0.39	4.6 (4.0)	22.9 (21.5)	–	0.60	–
KGW (context-hashed), matched, bias 2	0.745	0.17	0.11	0.07	0.12	4.4 (4.0)	23.2 (21.5)	–	0.56	–
dgMARK, greedy defaults	0.871	0.41	0.43	0.40	0.41	3.7 (3.3)	23.2 (59.4)	0.59	–	0.00
dgMARK, matched, top-k 3	0.983	0.84	0.78	0.64	0.78	8.9 (4.0)	34.2 (31.3)	0.70	–	0.44

D.4 Detection against perplexity at every strength

This subsection expands the comparison of Section 4.2. A single operating point per method can favor whichever method happens to sit at a lower perplexity. Table 12 therefore varies each method's strength under identical decoding, so that methods can be compared at matched perplexity. At similar Qwen perplexity the red–green list detects more deleted texts than TANGO. At perplexity 7.6 against TANGO's 7.2 it detects 96% against 68%, and at 9.9 against 8.4 it detects 98% against 81%.

Table 12: The red–green list detects more deleted texts than TANGO at matched perplexity. LLaDA, $n = 200$ per setting, with the 200 unwatermarked continuations as negatives. TANGO (either side) varies β, the red–green list its bias, and the Gumbel rule its sampling temperature. Perplexity is over non-degenerate continuations, which are at least 94% of every setting (Section B). The Gumbel row at $\tau = 1.0$ is a second generation run of the setting in Table 1, so it differs from that row by sampling noise.

setting	AUROC	clean	del30	syn30	ins20	PPL GPT-2	PPL Qwen
TANGO $\beta = 4$	0.998	0.97	0.61	0.84	0.85	16.8	6.1
TANGO $\beta = 5$ (Table 1)	0.998	0.97	0.68	0.92	0.91	19.4	7.2
TANGO $\beta = 6$	0.999	0.99	0.81	0.95	0.96	21.3	8.4
TANGO $\beta = 8$	1.000	1.00	0.93	0.99	0.98	31.4	13.5
TANGO $\beta = 10$	1.000	1.00	0.97	0.98	0.99	51.8	23.4
Red–green list, bias 2	0.940	0.58	0.38	0.39	0.69	13.5	4.6
Red–green list, bias 3	0.979	0.83	0.71	0.71	0.90	13.9	5.2
Red–green list, bias 4	0.995	0.94	0.85	0.85	0.96	16.3	6.3
Red–green list, bias 5 (Table 1)	0.993	0.98	0.96	0.94	0.98	18.0	7.6
Red–green list, bias 6	0.999	0.98	0.98	0.98	0.98	21.6	9.9
Gumbel $\tau = 1.0$	0.821	0.13	0.06	0.07	0.10	14.0	4.6
Gumbel $\tau = 1.2$	0.927	0.48	0.23	0.32	0.36	18.1	6.2
Gumbel $\tau = 1.5$	0.999	0.97	0.83	0.90	0.96	49.8	20.6

D.5 Results on Dream

This subsection expands the Dream rows of Table 1. On Dream the red–green list degrades text more than TANGO. Table 13 gives the Dream rows with their intervals and GPT-2 perplexity. The red–green list collapses 62% of its texts into a repeated phrase, against 20% of unwatermarked texts and 31% of TANGO texts (Section B). On the texts that do not collapse, its perplexity is also above TANGO's under both scorers. The collapse does not hurt its detection. Its mean score is 12.2 on the non-degenerate texts and 14.3 on the degenerate ones. The Dream null is in Section C, and Dream forgery from token pairs is in Table 9.

Key recovery from token frequencies stays at chance on Dream, as on LLaDA. For TANGO, the total-variation distance between watermarked and unwatermarked token frequencies is 0.319, above the 0.229 that sampling noise alone produces between two unwatermarked samples, and the red–green list reaches 0.610. With one-sided enforcement at $\beta = 8$ (100 texts), the key-recovery AUC is 0.503 against TANGO and 0.798 against the red–green list, and on the either-side texts of Table 1 it is 0.501 against TANGO. No forgery built from the predicted key passes TANGO's detector, while every one passes the red–green detector.

Table 13: On Dream, TANGO and the red–green list detect every unedited text, and TANGO has the lower perplexity on non-degenerate continuations. $n = 100$ per method, unwhitened coloring, with the 100 unwatermarked continuations as negatives and 95% bootstrap intervals. PPL is GPT-2-large perplexity over non-degenerate continuations, and *degenerate* is the fraction of continuations that collapsed into a repeated phrase (Section B).

		TPR@1%FPR					
method	AUROC	clean	del30	syn30	ins20	PPL GPT-2	degenerate
TANGO ($\beta = 5$, either side)	1.000	1.00 [1.00,1.00]	0.73 [0.65,0.81]	0.98 [0.95,1.00]	0.97 [0.93,1.00]	15.2	31%
Red–green list	1.000	1.00 [1.00,1.00]	0.98 [0.95,1.00]	0.98 [0.95,1.00]	0.98 [0.95,1.00]	18.1	62%
unwatermarked	–	–	–	–	–	9.9	20%

D.6 Removal by an attacker who holds the key

A provider whose key leaks should know how much editing removes the watermark. Table 14 compares two attackers who regenerate the same number of words. Because the attacker edits words and the checksum reads tokens, the key-aware attacker approximates each scored pair by the first tokens of two words that are two words apart. It uses the key to find the word pairs that satisfy the checksum and regenerates a random subset of those words. The random attacker regenerates random words. Both replace each chosen word with a prediction of a masked language model (distilroberta-base) that differs from the original word. The setting is one-sided enforcement at $\beta = 6$ with the whitened coloring ($n = 60$). Unedited texts have mean $z = 5.55$ at GPT-2 perplexity 19.2. At a 10% budget the two attackers lower the score by about the same amount (mean z 4.34 against 4.42). At 25% and 50% the key-aware attacker lowers it less than the random one, so for this attacker the key does not make removal easier. At a 25% budget about a quarter of the texts remain above $z = 4$, and at 50% almost none do. The key-aware attacker chooses which words to regenerate but not their replacements, and it locates taps at the word level while TANGO's tap is two tokens back. The table therefore describes only attackers of this kind.

Table 14: Knowing the key does not help this attacker remove the watermark. One-sided enforcement at $\beta = 6$, whitened coloring, $n = 60$, clean mean $z = 5.55$. Each budget is the fraction of words regenerated. *Det. rate* is the fraction of texts still above $z = 4$. PPL is GPT-2-large perplexity.

	key-aware			random		
budget	mean z	PPL GPT-2	det. rate	mean z	PPL GPT-2	det. rate
10%	4.34	28.1	0.62	4.42	28.5	0.60
25%	3.11	44.3	0.27	2.81	44.2	0.20
50%	1.54	76.8	0.02	1.03	95.2	0.02

D.7 Paraphrase

Theorem 2 covers edits that keep positions aligned. A paraphrase reorders whole clauses, so most tokens no longer sit at lag 2 from their original taps. We paraphrase by back-translation, from English to Chinese and back with the Helsinki-NLP Marian models ($n = 100$, one-sided enforcement at $\beta = 8$). TANGO's TPR@1%FPR falls from 1.00 to 0.34 (mean z from 6.5 to 0.9). The red–green list falls to 0.61 (mean z from 9.0 to 4.5), because its signal sits in single tokens and survives reordering. Neither watermark is robust to paraphrase.

D.8 Variance over seeds and keys

Single runs hide the variation due to the sampling seed and the secret key. Both checks below use one-sided settings, not the setting of Table 1. Table 15 repeats one-sided enforcement at $\beta = 8$ over three generation seeds (32 texts per seed). The per-seed standard deviation of TPR is 0.04 without edits and up to 0.19 under the heaviest edits. Table 16 repeats one-sided enforcement at $\beta = 6$ with a whitened coloring over four independent keys (40 texts per key). Clean AUROC is 0.998 ± 0.002 across keys, so detection does not depend on a lucky key direction.

Table 15: Detection over three generation seeds, one-sided enforcement at $\beta = 8$ with a single tap at lag 2 and the unwhitened coloring, 32 texts per seed, with the 96 unwatermarked continuations as negatives. Intervals are 95% percentile intervals from 1000 bootstrap resamples of the pooled texts.

attack	TPR@1%FPR (95% CI)	per-seed std
clean	0.98 [0.95,1.00]	0.04
del10	0.92 [0.85,0.97]	0.08
del30	0.62 [0.52,0.72]	0.18
syn30	0.85 [0.78,0.92]	0.11
syn50	0.61 [0.51,0.71]	0.19
ins20	0.91 [0.84,0.96]	0.09

Table 16: Detection over four independent secret keys, one-sided enforcement at $\beta = 6$, whitened coloring, 40 texts per key. The second row is the mean score of 300 natural C4 texts.

quantity	mean	std
clean AUROC	0.998	0.002
mean z on natural text	-0.11	0.28

E Full ablations

This appendix supports the design choices of Section 4.4. It gives every column of the ablations condensed into Table 2, the selection experiment that chose the enforcement side and bias, and a check of the length scaling of Proposition 2. The residue ablation is in Section C.3, next to the calibration diagnostic that explains it. Unless a caption says otherwise, the ablations use one-sided enforcement at $\beta = 8$ and residue $b = 0$, PPL is GPT-2-large perplexity over all continuations, and null FPR is the fraction of 200 natural C4 texts with $z > 4$.

The ablations of Tables 17 and 18 are small. Each cell has 24 watermarked texts, and the TPR@1%FPR threshold is set on the 24 unwatermarked continuations of the same prompts, so it is close to their maximum score. A cell at TPR 1.00 has an exact two-sided 95% Clopper–Pearson lower bound of 0.86. The noise is also visible directly. The $q = 3$ row of Table 17 and the $\{1\}$ row of Table 18 are the same configuration run twice, and their TPRs differ by up to 0.13. Differences of that size within a table are therefore not significant, and we draw conclusions only from trends across several rows.

E.1 Number of colors

The semantic-quantile coloring (*quantile* for short) keeps the null FPR at or below 0.5% for every number of colors, while the semantic-cluster coloring (*cluster*) of Section 3.1 does not (Table 17). The cluster coloring flags 60% of natural texts at $q = 2$ and 30.5% at $q = 3$, because tokens with similar embeddings, which often occur near each other, share a cluster, so the checksum holds by chance far more often than $1/q$. For the quantile coloring, $q = 2$ detects less under edits than $q \geq 3$, and perplexity grows from 22.2 at $q = 3$ to about 30 at $q \geq 8$.

Table 17: The semantic-quantile coloring keeps the null FPR at or below 0.5% for every q, while the cluster coloring flags 60% of natural texts at $q = 2$ and 30.5% at $q = 3$. Lag 1, $n = 24$ per row.

coloring	AUROC	clean	del10	del30	syn30	syn50	ins20	PPL GPT-2	null FPR (%)
quantile, $q = 2$	0.989	0.92	0.79	0.62	0.71	0.58	0.83	20.4	0.0
quantile, $q = 3$	1.000	1.00	1.00	0.92	0.96	0.75	1.00	22.2	0.0
quantile, $q = 4$	1.000	1.00	0.96	0.79	0.83	0.67	0.96	26.8	0.5
quantile, $q = 8$	1.000	1.00	1.00	0.83	0.83	0.62	1.00	29.9	0.5
quantile, $q = 16$	0.998	0.96	0.96	0.88	0.96	0.92	0.96	29.5	0.0
cluster, $q = 2$	0.817	0.38	0.25	0.12	0.17	0.08	0.46	13.3	60.0
cluster, $q = 3$	0.916	0.71	0.58	0.46	0.54	0.50	0.54	17.5	30.5
cluster, $q = 4$	0.972	0.92	0.92	0.92	0.83	0.75	0.92	23.3	0.0
cluster, $q = 8$	0.930	0.67	0.75	0.67	0.83	0.79	0.62	24.0	0.0
cluster, $q = 16$	0.975	0.79	0.75	0.58	0.50	0.42	0.75	20.3	0.5

E.2 Tap set

Detection under edits falls as taps are added (Table 18 and Figure 3). This is consistent with Theorem 2, by which each added tap multiplies the expected score under edits by another factor $1 - \rho$. A single tap detects best under every edit, and with four taps del30 TPR falls from 0.92 to 0.12. The trend is not monotone in every column, but the exceptions are within the noise of 24 texts per row. The theorem depends only on $|\mathcal{D}|$, so it predicts the same robustness for lag 1 and lag 2. In the data, lag 2 detects as well as or better than lag 1 in every column. We use lag 2 because the adjacent token's color is the most correlated with the scored token's color in natural text (Section C.3), and because the lag-1 tap leaks more to the pair-counting attacker of Table 9 (Table 19).

Table 18: Adding taps lowers detection under edits. $q = 3$ semantic-quantile coloring, $n = 24$ per row.

taps $\mathcal{D}$	AUROC	clean	del10	del30	syn30	syn50	ins20	PPL GPT-2	null FPR (%)
$\{1\}$	0.990	0.96	0.96	0.92	0.83	0.71	0.96	25.6	0.0
$\{2\}$	1.000	1.00	1.00	0.92	0.96	0.92	0.96	25.0	0.0
$\{1, 2\}$	0.990	0.96	0.79	0.25	0.58	0.17	0.79	19.6	0.0
$\{1, 3\}$	1.000	1.00	0.88	0.50	0.62	0.62	0.75	18.5	0.0
$\{1, 2, 3\}$	0.986	0.83	0.62	0.25	0.25	0.04	0.54	16.7	0.0
$\{1, 2, 3, 4\}$	0.984	0.83	0.50	0.12	0.38	0.25	0.42	18.9	0.0

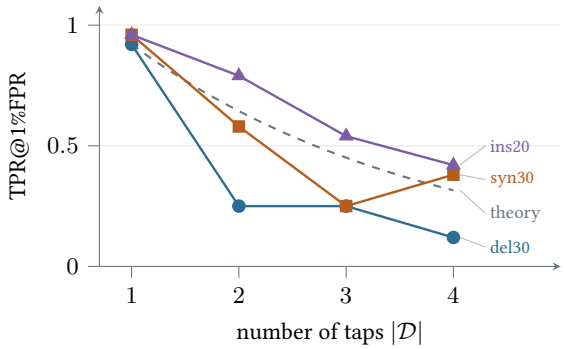

Figure 3: Detection under edits falls as taps are added (rows $\{2\}$, $\{1,2\}$, $\{1,2,3\}$, and $\{1,2,3,4\}$ of Table 18). The dashed curve scales the single-tap del30 TPR by $(1-\rho)^{|\mathcal{D}|-1}$ at $\rho = 0.3$. It shows only the trend that Theorem 2 predicts, because the theorem concerns the mean score under substitution.

E.3 Enforcement side, bias, lag, and unmasking order

Either-side enforcement raises detection under deletion, and at $\beta = 6$ it costs little Qwen perplexity (7.8 against 7.4). The other variants of the selection experiment each cost more perplexity or leak more to forgery (Table 19, rule in Section B). Besides the fixed bias β, the experiment tried a *target* bias, which at each enforced position adds the smallest bias, up to a cap, that lifts the favored class to a target probability. It also tried a lag-1 tap and *order steering*. Order steering targets positions whose pair cannot yet be enforced, because neither neighbor at lag δ is unmasked. It halves the confidence score that the sampler uses to rank these positions for unmasking. The sampler then tends to unmask positions with an unmasked neighbor first.

Enforcing from either side raises the enforced fraction from 0.61 to 0.74 at $\beta = 8$ and from 0.62 to 0.75 at $\beta = 6$, and it raises del30 TPR at both biases. A lag-1 tap reaches del30 TPR 0.94, against 0.77 for lag 2 at the same $\beta = 6$ and perplexity, but its forgery success is 0.54 against 0.38. Order steering raises the enforced fraction to about 0.9, but across the three steered settings it multiplies perplexity by 1.3 to 1.7 (Qwen and GPT-2), and at $\beta = 6$ it lowers AUROC to 0.974.

Table 19: Either-side enforcement at $\beta = 5$ (bold) has the lowest Qwen perplexity among the eligible settings of the selection experiment. 48 held-out C4 prompts, with their 48 unwatermarked continuations as negatives. *Pair AUC / forge* is the attack of Table 9 at $N = 48$. *Target, cap* biases each enforced position just enough to lift the favored class to the target probability, with a bias of at most the cap. *Steered* is order steering. ○ marks eligible settings, × marks eligible settings excluded for forgery, and unmarked settings are ineligible (rule in Section B). Perplexity is over all continuations.

setting		AUROC	clean	del30	syn30	ins20	enforced fraction	PPL GPT-2	PPL Qwen	pair AUC / forge
one-sided, $\beta = 8$ (reference)		0.999	0.98	0.79	0.98	0.98	0.61	24.0	9.7	0.61 / 0.36
one-sided, $\beta = 6$		0.999	0.98	0.75	0.79	0.96	0.62	19.6	7.4	0.61 / 0.26
either side, $\beta = 8$	×	1.000	1.00	0.94	1.00	1.00	0.74	31.1	12.7	0.62 / 0.47
either side, $\beta = 6$		1.000	1.00	0.77	0.96	0.96	0.75	19.6	7.8	0.63 / 0.38
either side, $\beta = 5$	○	**1.000**	**1.00**	**0.79**	**0.92**	**0.98**	**0.76**	**17.8**	**7.0**	**0.62 / 0.40**
either side, $\beta = 4$		0.998	0.96	0.73	0.81	0.81	0.77	16.0	5.9	0.61 / 0.23
either side, target 0.8, cap 8	○	1.000	1.00	0.92	0.96	0.98	0.71	25.7	11.0	0.62 / 0.42
either side, target 0.9, cap 8	○	1.000	1.00	0.92	1.00	1.00	0.72	27.8	11.7	0.62 / 0.34
either side, target 0.95, cap 8	×	1.000	1.00	0.88	1.00	1.00	0.73	29.0	11.9	0.63 / 0.49
either side, target 0.9, cap 6	○	1.000	1.00	0.79	0.83	0.92	0.73	19.8	8.0	0.62 / 0.28
either side, $\beta = 6$, lag 1	×	1.000	1.00	0.94	0.98	0.98	0.73	19.6	7.6	0.65 / 0.54
either side, $\beta = 6$, steered		0.974	0.96	0.85	0.94	0.94	0.90	33.9	11.2	0.61 / 0.40
either side, $\beta = 8$, steered		0.979	0.98	0.96	0.98	0.98	0.88	44.9	21.2	0.62 / 0.62
either side, target 0.9, cap 8, steered	×	1.000	1.00	0.96	1.00	1.00	0.86	36.3	17.8	0.63 / 0.57
unwatermarked								12.1	4.2	

E.4 Detection grows with text length

Proposition 2 predicts that the mean score grows as $\sqrt{T}$ in the number T of scored pairs, with slope ε/σ_0 when the margin ε does not depend on length. Figure 4 checks the $\sqrt{T}$ shape on 200 LLaDA texts. We truncate each 128-token text to its first 16, 32, 48, 64, 96, or 128 tokens and score each prefix. Some texts end early, so the full texts have 124.9 scored pairs on average. The slope 0.48 comes from the margin measured on the full texts ($\hat{\varepsilon} = 0.227$), so the line and the data agree at full length by construction. The test is whether shorter prefixes fall on the same line. At 94 scored pairs the measured mean score matches the line (4.67 in both cases). For shorter prefixes it is lower, so the margin is smaller near the start of a text than over the whole text. The line reaches $z = 4$ at $T \approx 69$ scored pairs, and the measured scores reach it at about $T = 78$. The prediction therefore underestimates the length of text that detection needs by about 9 scored pairs.

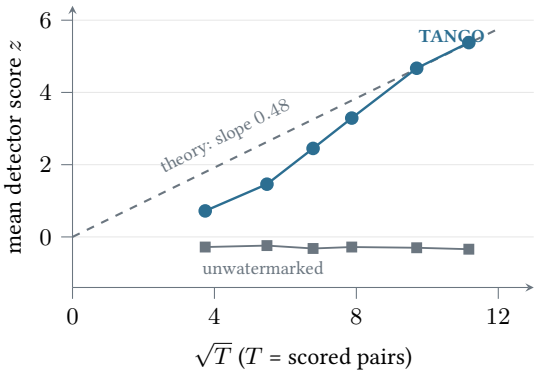

Figure 4: The mean score matches the predicted line at 94 scored pairs and falls below it for shorter prefixes. Unwatermarked text stays near zero. Prefixes of 200 LLaDA texts, one-sided enforcement at $\beta = 6$, whitened coloring. The dashed line has slope $\hat{\varepsilon}/\sigma_0 = 0.48$, with $\hat{\varepsilon}$ measured on the full texts.

F Additional related work

This appendix extends Section 5 with further watermarks for diffusion language models, semantic watermarks, and attacks and defenses around forgery.

Further watermarks for diffusion language models. DMark (Wu et al., 2025) extends the context-hashed red–green bias of Gloaguen et al. (2026) with predicted and bidirectional context. Bagchi et al. (2025) apply Gumbel-max sampling at every diffusion step, seeded by position, and SAC-Copula (Li & He, 2026) replaces its independent perturbations with locally correlated Gumbel fields. Zhao (2026) controls a global sketch of the whole sequence, which gives an order-agnostic statistic. DenMark (Ma et al., 2026), concurrent with our work, embeds a semantic watermark in the denoising process.

Semantic watermarks. Watermarks keyed by hashes of tokens break under paraphrase, which motivated watermarks keyed by meaning. SIR derives the watermark logits from an embedding of the preceding context (Liu et al., 2024b), SemStamp partitions a sentence-embedding space and rejection-samples into keyed regions (Hou et al., 2024), and SemaMark discretizes context semantics (Ren et al., 2024). These methods use meaning to make the key assignment of a single token or sentence robust to paraphrase. TANGO uses meaning to make a pair statistic survive synonym substitution. Like most semantic watermarks, it needs an embedding matrix at detection. Image watermarks for diffusion models share both ideas, embedding the mark inside the sampling process and keying it to semantics. Hidden-in-the-Noise places the mark in the initial diffusion noise (Arabi et al., 2025a), and SEAL keys it to image semantics (Arabi et al., 2025b).

Stealing, spoofing, and signatures. Watermark stealing is practical. With a modest number of texts an attacker approximates the green list well enough to spoof or remove the watermark

(Jovanović et al., 2024), and random-selection probing recovers the red–green lists of n-gram watermarks (Chen et al., 2024). Comparing output token frequencies also reveals token colors (Wu & Chandrasekaran, 2024), and Zhang et al. (2024b) recover the green list of a single-key scheme with mixed integer programming and use it to remove the watermark. A student model can also learn to generate watermarked text by distillation, which lets an attacker spoof a watermark without reading its key (Gu et al., 2024). Spoofing was an early objection to detector-based provenance (Sadasivan et al., 2025), and Zhang et al. (2024a) show that strong watermarking is impossible against an attacker with a quality oracle and a perturbation oracle. Defenses include statistical tests that flag learning-based spoofing (Gloaguen et al., 2024), and Zhao et al. (2025) list robust, unforgeable public attribution among the open problems of watermarking. Liu et al. (2024a) use separate generation and detection networks so that the public detector is hard to forge from, and Bileve embeds signature bits to detect spoofing (Zhou et al., 2024). The cryptographic response embeds a digital signature in the text (Fairoze et al., 2025), and Lin et al. (2026) make the signature robust to token substitutions. A signature's guarantee is computational and holds against any computationally bounded attacker. TANGO's guarantee is statistical. Under balanced colors and an evenly varying favored class, expected token frequencies carry no information about the key (Theorem 1), and the guarantee says nothing about token pairs. Signature schemes require rejection sampling over high-entropy blocks. TANGO adds one logit update per position, and its expected score decreases at a known rate under random substitutions (Theorem 2).

G Qualitative samples

Figure 5 shows continuations of three C4 prompts by the unwatermarked model, TANGO, the red–green list, and the Gumbel rule at the settings of Table 1. Each entry is one or two sentences copied verbatim from the full continuation. In these samples all four methods produce readable text, and TANGO's text shows no visible trace of the checksum.

Prompt. … enjoying the sounds of holiday music, or taking a trip back in time to celebrate the traditions

UNWATERMARKED. Each experience enriches the holiday season, making it a time of personalization and cherished memories.

TANGO. Exchanging a wish list with Santa brings a sense of anticipation and joy, while the sounds of holiday music can transport you straight to the festive season.

RED–GREEN LIST. It's a time to reunite with loved ones, create cherished memories, and appreciate the small joys that make the holiday season truly special.

GUMBEL. Sharing your wish list with Santa can spark a sense of hope and anticipation, while the enchanting sounds of holiday music can transport you to cozy fireplaces and starlit nights.

Prompt. Students have long applied to colleges and universities with applications that are heavy on test scores and grades … the founders of

UNWATERMARKED. However, over time, standardized test scores like the SAT and ACT (or A Levels in the UK) and academic grades have become significant components of the admissions process.

TANGO. Many founders recognize that reliance on test scores and grades does not always reflect a student's overall potential, character, or diverse interests.

RED–GREEN LIST. While that's not necessarily wrong, the founders of colleges often emphasized the importance of considering a student's overall character, extracurricular activities, personal statements, letters of recommendation, and other elements that demonstrate well-roundedness.

GUMBEL. Historically, founders wanted to assess a student's potential for academic success, extracurricular activities, personal qualities, and other factors.

Prompt. Cluster comprises IBM's Opteron-based eServer 325 server and systems management software and storage devices that can run Linux and Windows operating systems

UNWATERMARKED. This setup is ideal for heterogeneous computing environments where multiple operating systems need to coexist and interact seamlessly.

TANGO. IBM's eServer 325 servers are known for their efficient performance and reliability, making them suitable for various computational needs.

RED–GREEN LIST. This configuration allows for a high performance, scalable, and compatible server architecture that can accommodate various operating systems and applications.

GUMBEL. This setup allows for flexibility and scalability, as it can run multiple operating systems concurrently, which can be beneficial for applications that require heterogeneous environments.

Figure 5: Continuations of three C4 prompts at the settings of Table 1, each a verbatim span of the full continuation.